Enterprise Application Security as a Quality Issue

- *"Over 70 percent of security vulnerabilities exist at the application layer, not the network layer"* (Gartner)
- *"Hacking has moved from a hobbyist pursuit with a goal of notoriety to a criminal pursuit with a goal of money"* (Counterpane Internet Security)
- *"64 percent of developers are not confident in their ability to write secure applications"* (Microsoft Developer Research)
- *"Losses arising from vulnerable web applications are significant and expensive – up to $60 billion annually"* (IDC/IBM Systems Sciences Institute)
- *"If 50 percent of software vulnerabilities were removed prior to production use, enterprise configuration management and incident response costs would be reduced by 75 percent each."* (Gartner)

iqnite GERMANY2010

No quality without security – SOA Security is more than Web Service Testing

Dr. Alexander Schinner, CISSP, GCFA, GCIA & Dominik Kopriva

T Systems

Agenda

- Services and Web Services
- Web Services and Security
- Security and Testing
- Protect and Attack

Services and Web Services

Introduction to SOA
Service-Oriented Architecture

- Principal characteristics
  - architectural style
  - composed from loosely coupled services
  - interaction by sending messages
  - independent services
  - service description defines
    - interface
    - constraints
    - policies that must be respected
  - a service is a building block
- An implementation platform is the Web Services technology

Introduction to SOA

Multiple Players

„Service Oriented Architecture is a paradigm for organizing and utilizing distributed capabilites that may be under the control of different ownership domains“

- Services
  - black boxes
  - platform independent
  - interoperable
  - (idempotent)
  - autonomic
  - reusable
  - loose coupling
  - stateless
- Message Pattern
  - Request – Response
  - Publish – Subscribe
- Techniques
  - UDDI
  - WSDL
  - BPEL
  - SOAP

Introduction to Security
CIA

- Confidentiality

  Information about system or its users cannot be learned by an attacker

Introduction to Security
CIA

- Confidentiality

  Information about system or its users cannot be learned by an attacker

- Integrity

  The system continues to operate properly, only reaching states that would occur if there were no attacker

Introduction to Security
CIA

- Confidentiality

  Information about system or its users cannot be learned by an attacker

- Integrity

  The system continues to operate properly, only reaching states that would occur if there were no attacker

- Availability

  Actions by an attacker do not prevent users from having access to use of the system

Introduction to Security

Impact of Damage[1]

Potential impact is LOW if:

- The loss of confidentiality, integrity, or availability could be expected to have a limited adverse effect on organizational operations, assets, or individuals.

Potential impact is MODERATE if:

- The loss of confidentiality, integrity, or availability could be expected to have a serious adverse effect on organizational operations, assets, or individuals.

Potential impact is HIGH if:

- The loss of confidentiality, integrity, or availability could be expected to have a severe or catastrophic adverse effect on organizational operations, assets, or individuals.

[1]Adapted from FIPS Publication 199: Standards for Security Categorization of Federal Information and Information Systems

Introduction to SOA

Security Point of View

T Systems

Introduction to SOA

Security Point of View

SOA

- System correctness

  Good input $\Rightarrow$ Good output

Security

- Security

  Bad input $\not\Rightarrow$ Bad output

Introduction to SOA

Security Point of View

SOA

- System correctness

  Good input $\Rightarrow$ Good output

- System correctness

  More features: better

Security

- Security

  Bad input $\not\Rightarrow$ Bad output

- Security

  More features: can be worse

Introduction to SOA

Security Point of View

- A service edge is a natural boundary

SOA

Warning

"Remote Calls" hide the presence of the networks and boundary.

→This is evil ™

- System correctr

  Good input ⇒ Good output

- Security

  Bad input ⇏ Bad output

- System correctness

  More features: better

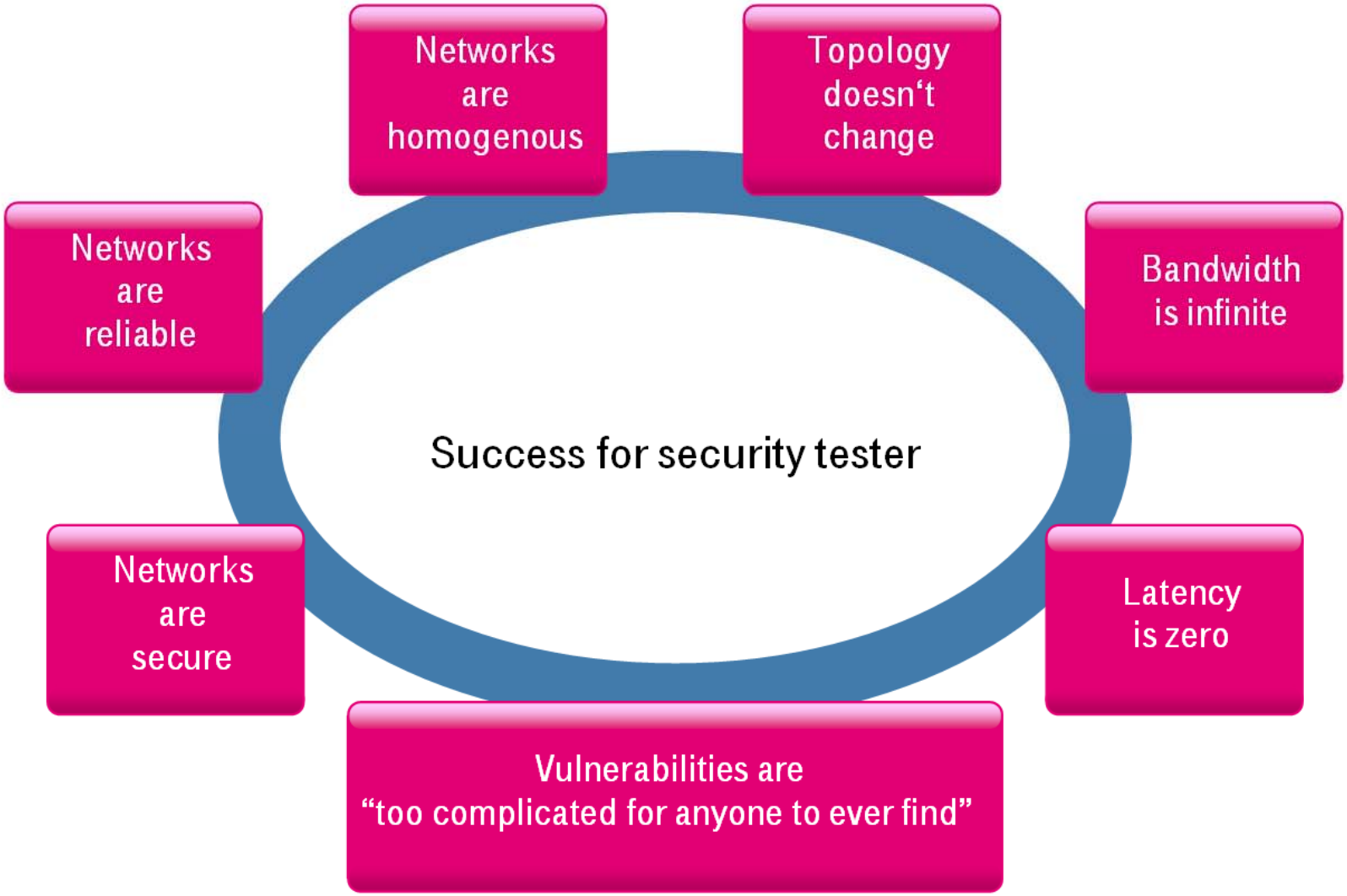
- Security

  More features: can be worse

T Systems

Introduction to SOA

Fallacies of distributed computing



Introduction to SOA Security

Why Computer Security

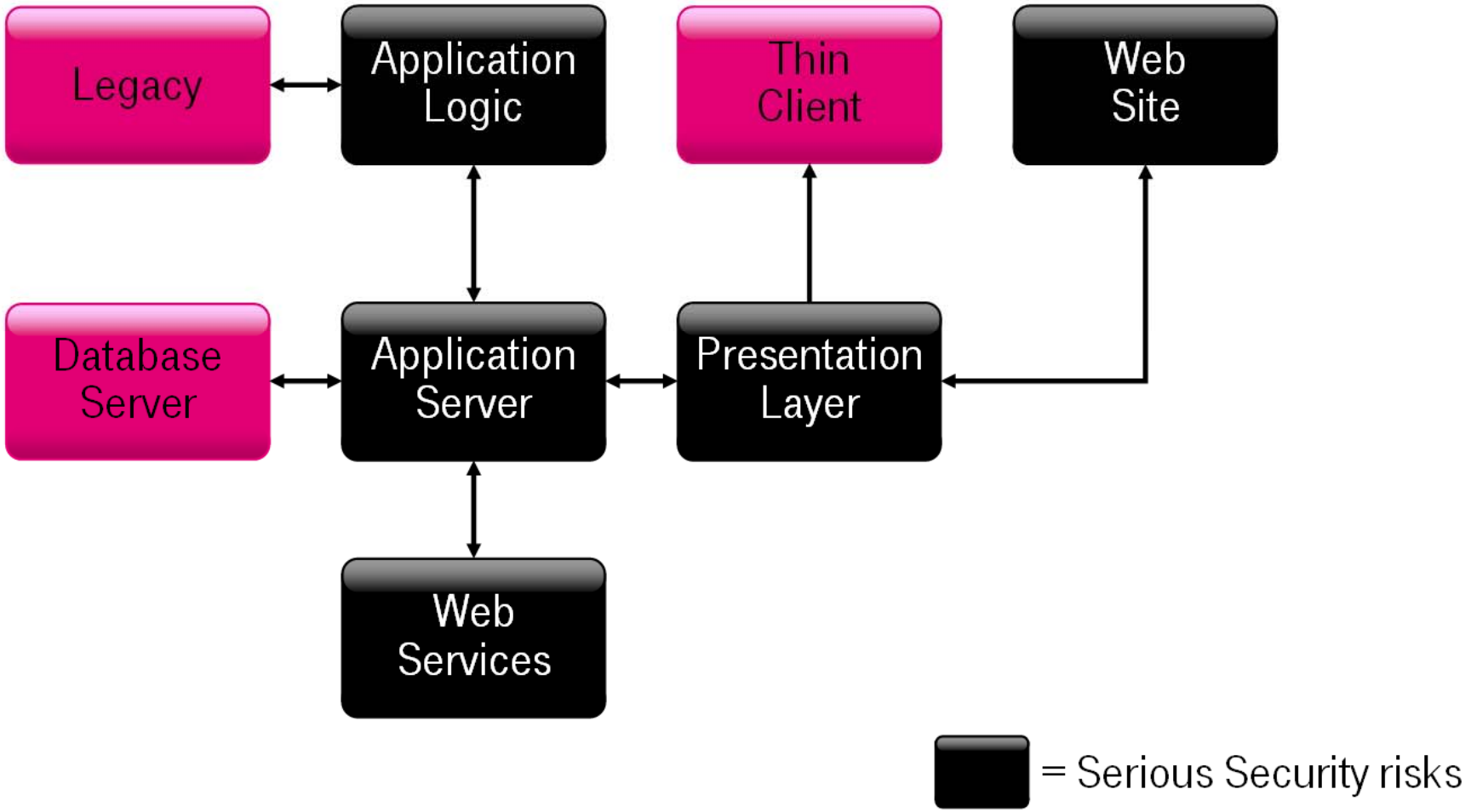
- The Internet and Intranet is a dangerous place
- Business data is an attractive target
- Need to protect
  - data
  - ability to use our business
  - reputation with customer and the general public

- Areas of Concern
  - Organizational operations
    - Mission capability to perform its primary functions, with effectiveness
  - Organizational assets: damage or loss
  - Financial loss
  - Individuals

Web Services and Security

Web Service Security

Security Challenges



Web Service Security

Tasks to fulfill

1. Authentication - Identify who you are
   - Userid/password, PKI certificates, Kerberos, Tokens, Biometrics
2. Authorization – What you can access
   - Access Enforcement Function / Access Decision Function
   - Roles, Groups, Entitlements
3. Administration – Applying security policy to resource protection
   - Directories, administration interfaces, delegation, self-service
4. Audit – Logging security success / failures
   - Basis of monitoring, accountability/non-repudiation, investigation, forensics
5. Assurance – Security integrity and compliance to policy
   - Monitoring (Intrusion Detection, AntiVirus, Compliance), Vulnerability Testing
6. Asset Protection
   - Data Confidentiality, Integrity, Data Privacy
7. Availability
   - Backup/recovery, disaster recovery, high availability/redundance

SOA Security Challenges 1

Problems with Web Services and SOA

- Cut through firewall
  - SOAP messages often travel over HTTP port 80
- Business processes on the web
  - Expose internal APIs to anonymous users
- New technology, new mistakes
  - Once web apps are locked tighter, guess who's next?
- Implied assumptions, external dependence
  - "I can't see it, neither can a hacker"
  - "We can trust that service to work properly"
  - "The use of the service is constrained by the client application"

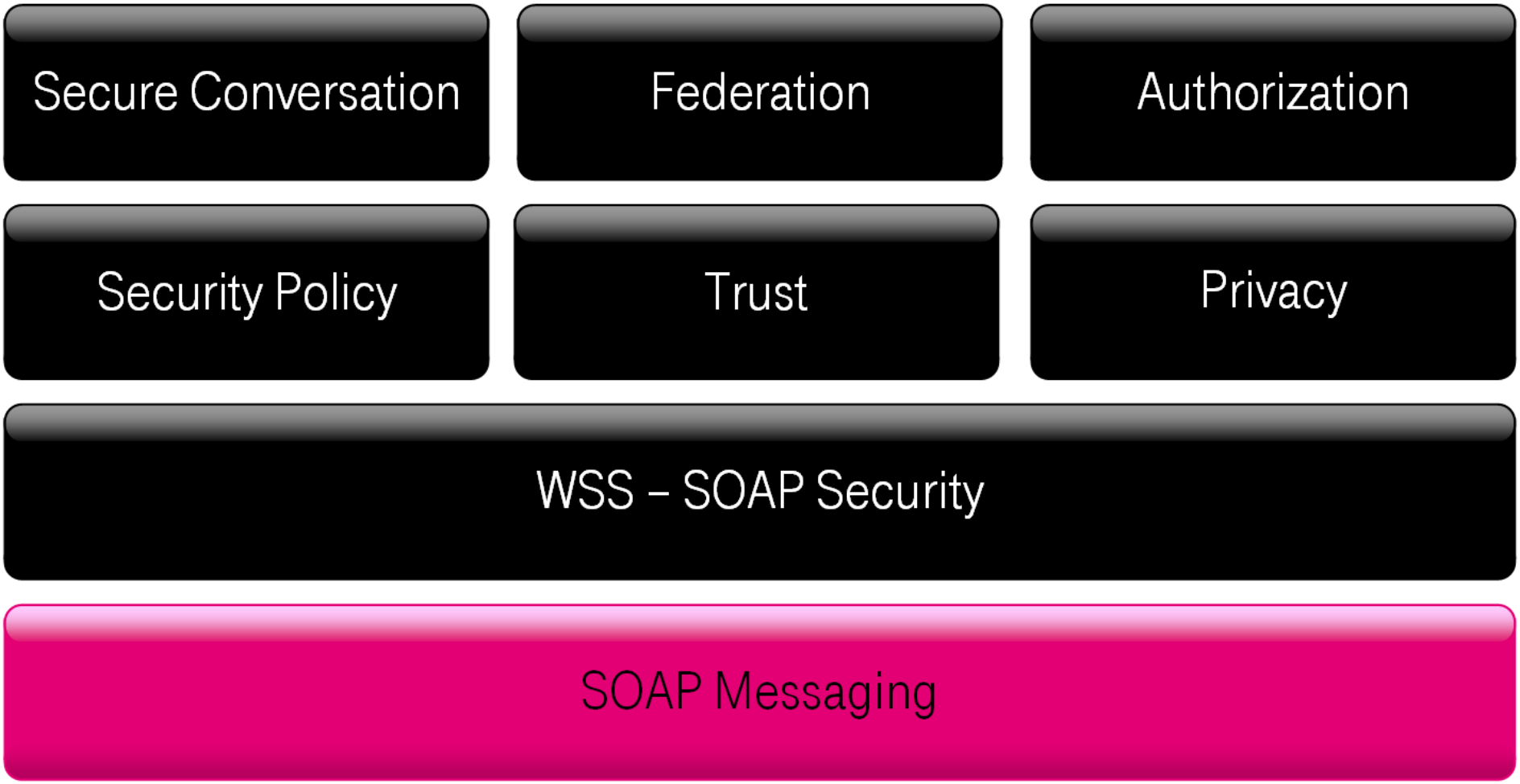
SOA Security Challenges 2

The Year 2000 problem revisited

- Former Days: Y2K problems
  - Applications never designed to work > 30,40 years
  - Source code contains principal cause of the problems
  - One defect (bug) is enough to cause serious problems
- Today: SOA
  - Applications never designed to be connected to networks / internet
  - Source code contains root cause of the problems
  - One vulnerability is enough to cause serious risk
  - And worse, people hunt for them !

Web Service Security

Specifications Roadmap

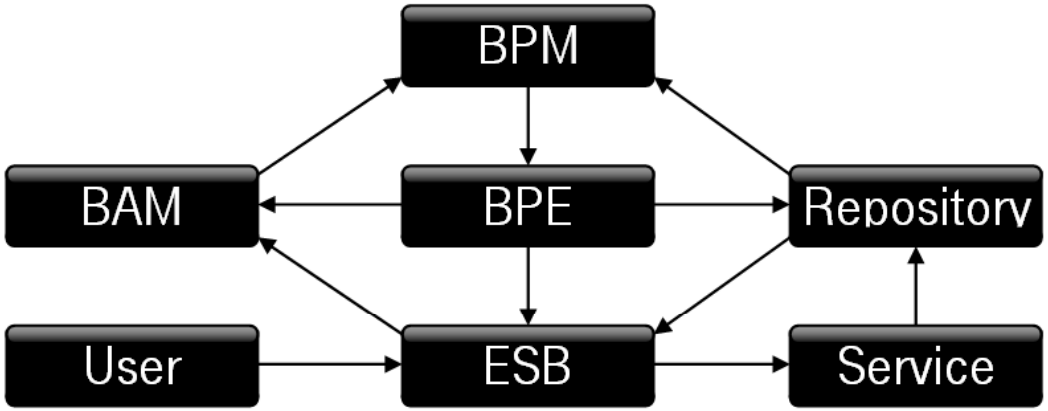


T Systems

Service Oriented Architecture

Attack Scenarios - SOA Components

- Business Activity Monitor
- Business Process Monitor
- Business Process Engine
- Repository
- Service Provider
- Enterprise Service Bus
- Service User



The “Service” is the “Holy Grail” for the attacker, but...

T Systems

Attackers Point of View

- Every step will help the attacker!

Attackers Point of View

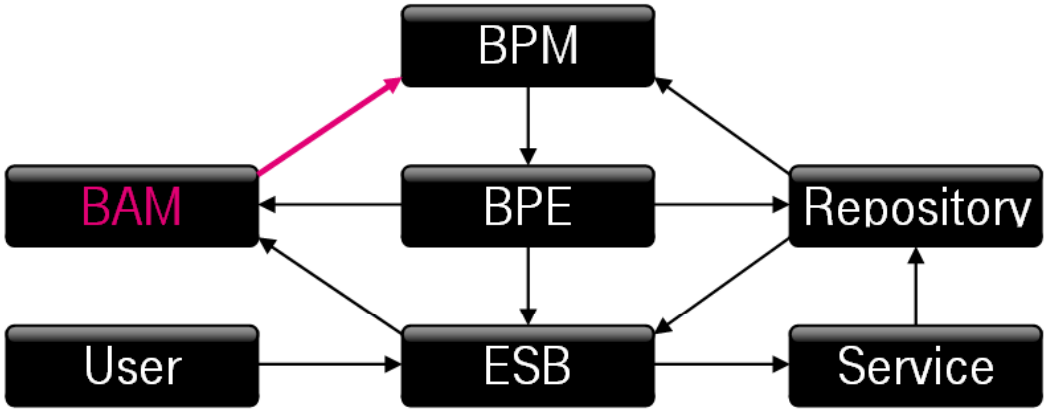
- Every step will help the attacker!

T Systems

Service Oriented Architecture

Attack Scenarios - Business Activity Monitor

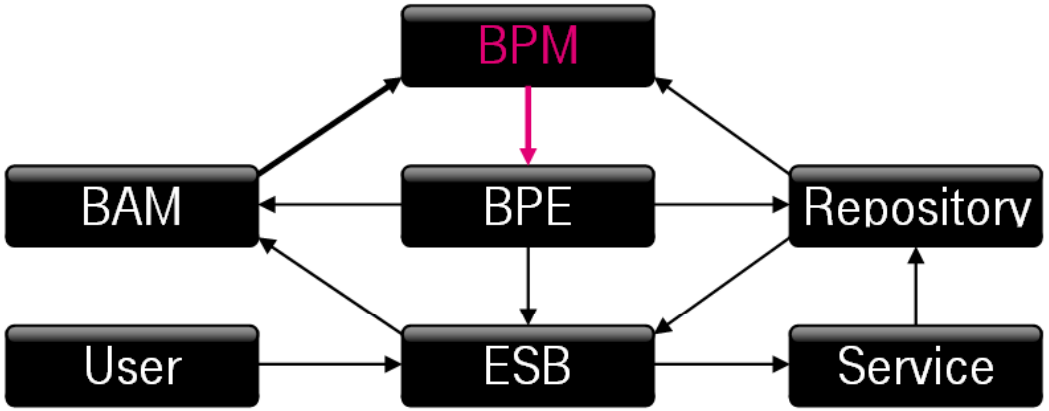
- Target
  - Business Process Monitor
- Attack Vectors
  - Manipulate Monitoring
  - Trigger action
- Risk
  - Low
  - Medium



Service Oriented Architecture

Attack Scenarios - Business Process Monitor

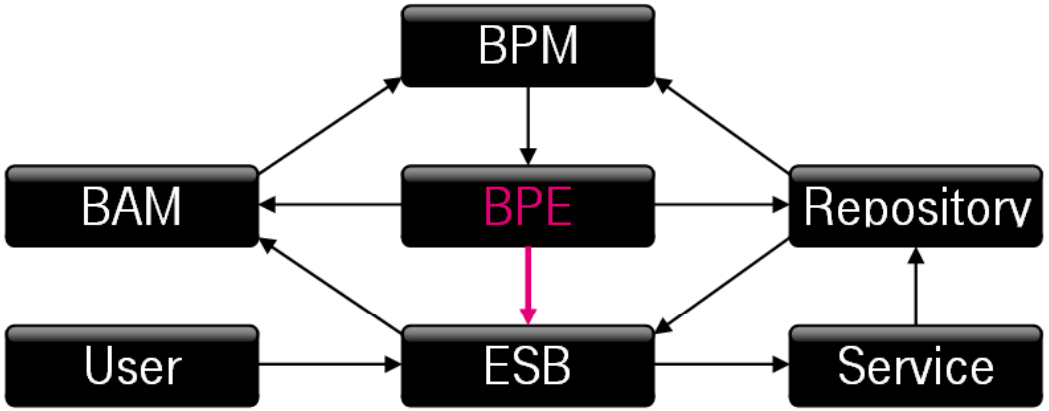
- Target
  - Business Process Engine
- Attack Vectors
  - Manipulate Processes
  - Manipulate Monitoring
- Risk
  - Medium
  - High



Service Oriented Architecture

Attack Scenarios - Business Process Engine

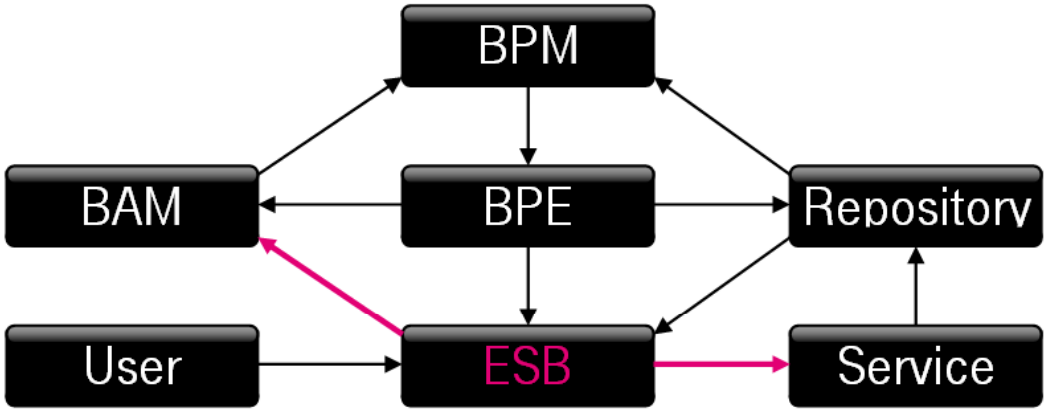
- Target
  - Enterprise Service Bus
  - Brain of the system
- Attack Vectors
  - Manipulate Processes
  - Manipulate Transport
- Risk
  - High



Service Oriented Architecture

Attack Scenarios - Enterprise Service Bus

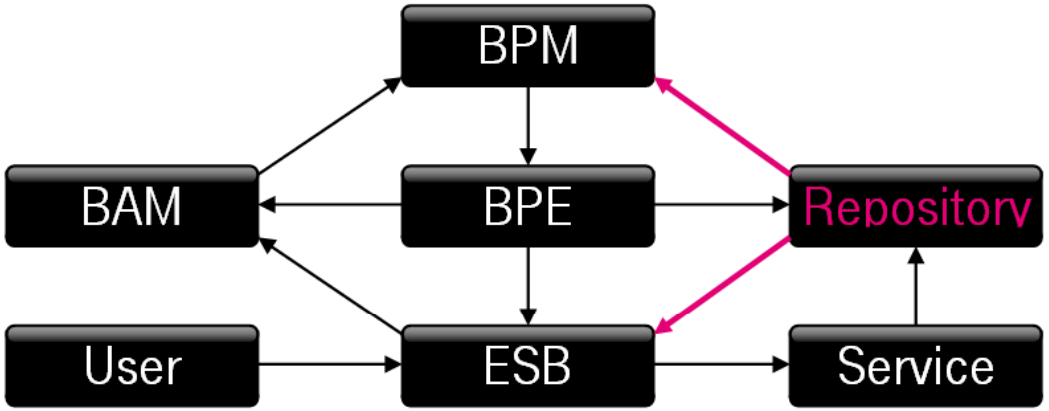
- Target
  - Service provider
  - Business Activity Monitor
  - Backbone of the system
- Attack Vectors
  - Manipulate Processes
  - Manipulate Monitoring
  - Attack and manipulate Services
- Risk
  - High



Service Oriented Architecture

Attack Scenarios - Repository

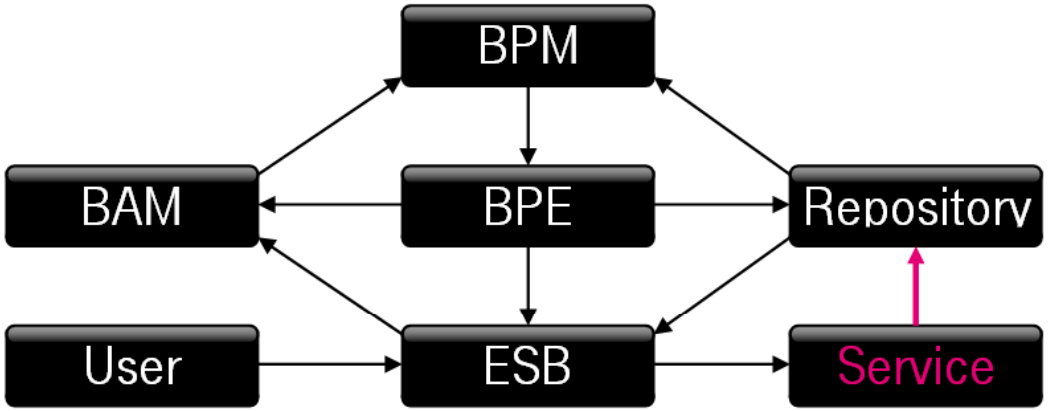
- Target
  - Business Process Monitor
  - Enterprise Service Bus
- Attack Vectors
  - Manipulate orchestration
  - Manipulate data transport
  - Manipulate service information
- Risk
  - Medium
  - High



Service Oriented Architecture

Attack Scenarios – Service Provider

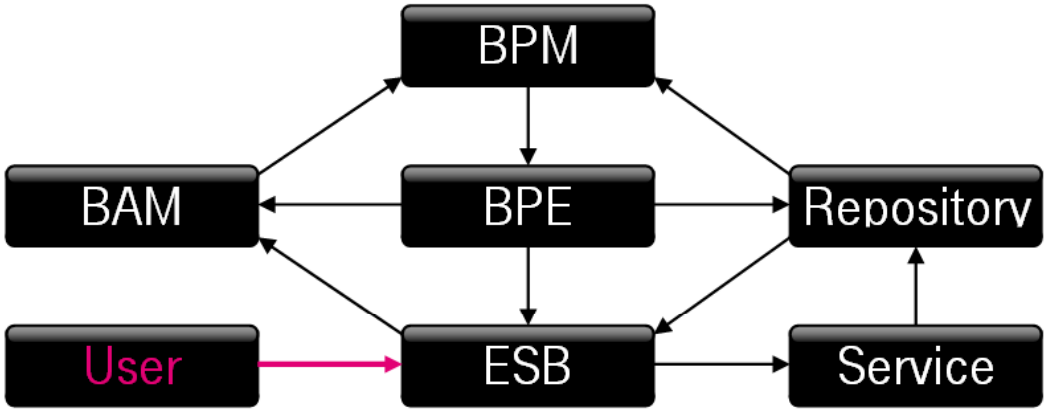
- Target
  - Repository
- Attack Vectors
  - Manipulate Repository information
  - everything else...
- Risk
  - High



Service Oriented Architecture

Attack Scenarios - User

- Target
  - Enterprise Service Bus
  - Weakest target
- Attack Vectors
  - Manipulate Processes
  - Manipulate Monitoring
- Risk
  - Low
  - Medium
  - High



Security and Testing

Web Service Security

SOA Testing Challenges

Test Automation

- Client Simulators (Message simulation)
- Service Simulators
- Automatic test data creation
- Choosing the right test automation tool

Major Challenges

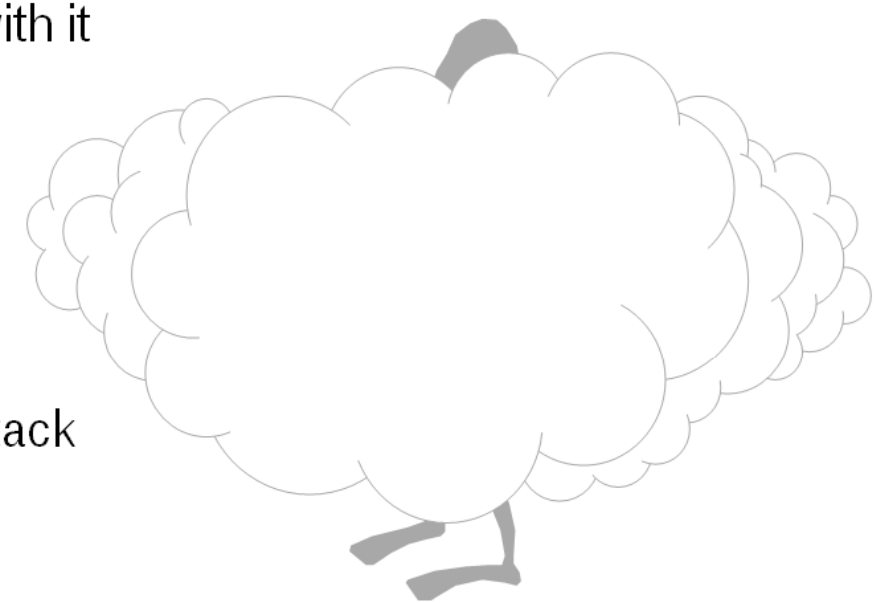
- No user interface for the Web Services
- Difficult to create Real time applications
- Test Data should be close to the end user environment
- Infinite consumers possible

Maintaining pool of testing resources with SOA domain knowledge

Web Service Security

SOA Vulnerabilities

- Web Services vulnerabilities can be present in the:
  - Operating system or the applications that ship with it
  - Network
  - Database
  - Web server
  - Application server
  - XML parser or Web services implementation / stack
  - Application code
  - XML appliance



Web Service Security

Example XML Bomb

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE SOAP-ENV:Envelope [
  <!ELEMENT SOAP-ENV:Envelope ANY>
  <!ATTLIST SOAP-ENV:Envelope entityReference CDATA #IMPLIED>
  <!ENTITY x0 "Bomb!">
  <!ENTITY x1 "&x0;&x0;">
  <!ENTITY x2 "&x1;&x1;">
  ...
  <!ENTITY x20 "&x19;&x19;">
  <!ENTITY x21 "&x20;&x20;">
  …
  <!ENTITY x99 "&x98;&x98;">
]>
```

Web Service Security

Example XML Bomb

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE SOAP-ENV:Envelope [
  <!ELEMENT SOAP-ENV:Envelope ANY>
  <!ATTLIST SOAP-ENV:Envelope entityReference CDATA #IMPLIED>
  <!ENTITY x0 "Bomb!">
  <!ENTITY x1 "&x0;&x0;">
  <!ENTITY x2 "&x1;&x1;">
  ...
  <!ENTITY x20 "&x19;&x19;">
  <!ENTITY x21 "&x20;&x20;">
  …
  <!ENTITY x99 "&x98;&x98;">
]>
```

2^{99} = 633825300114114700748351602688

Web Service Security

General Web Services Threats

Common to all Web applications

- SQL Injections
  - Special characters in queries
- Capture and Replay Attacks
  - Man in the middle attacks
- DoS (resulting from a large load)
  - Blow up application from inside
- Improper Error Handling
  - Dump of stack trace etc
- Broken Access Control
  - Take over earlier sessions tokens etc

Specific to XML Web services

- Large Payloads
  - Send huge XML load, or generate huge responses
- XPath Injections
  - Query XML documents for certain nodes
- External Entity Attacks
  - Misuse pointed to XML data using URI
- XML Bombs
  - Recursive XML entity declaration

Web Service Security

General Web Services Threats Prevented

- SQL Injections

Policy

- Validate user input
- strip potentially malicious characters like ‘ and “ as soon as you get the

Test

- Penetrate
- Regression test

Web Service Security

General Web Services Threats Prevented

- SQL Injections
- Capture and Replay Attacks

Policy

- Use signed random nonce values and Timestamps

Test

- Penetrate
- Regression test

Web Service Security

General Web Services Threats Prevented

- SQL Injections
- Capture and Replay Attacks
- DoS (resulting from a large load)

Policy

- Secure coding standards

Test

- Simulate attacks
- Regression test

Web Service Security

General Web Services Threats Prevented

- SQL Injections
- Capture and Replay Attacks
- DoS (resulting from a large load)
- Improper Error Handling

Policy

- Catch/handle all exceptions
- Secure coding standards

Test

- Penetrate
- Regression test

Web Service Security

General Web Services Threats Prevented

- SQL Injections
- Capture and Replay Attacks
- DoS (resulting from a large load)
- Improper Error Handling
- Broken Access Control

Policy

- Baseline security policies
- Extended security policies

Test

- Penetrate
- Positive & negative conditions
- regression test

T Systems

Web Service Security

General Web Services Threats Prevented

- SQL Injections
- Capture and Replay Attacks
- DoS (resulting from a large load)
- Improper Error Handling
- Broken Access Control
- Large Payloads

Policy

- Validate input
- Constrain schema types

Test

- Simulate attacks
- regression test

Web Service Security

General Web Services Threats Prevented

- SQL Injections
- Capture and Replay Attacks
- DoS (resulting from a large load)
- Improper Error Handling
- Broken Access Control
- Large Payloads
- XPath Injections

Policy

- Validate user input
- strip potentially malicious characters like ‘ and “ as soon as you get the

Test

- Simulate attacks
- Regression test

T Systems

Web Service Security

General Web Services Threats Prevented

- SQL Injections
- Capture and Replay Attacks
- DoS (resulting from a large load)
- Improper Error Handling
- Broken Access Control
- Large Payloads
- XPath Injections
- External Entity Attacks

Policy

- Disable DTD processing in XML parser

Test

- Simulate attacks
- Regression test

Web Service Security

General Web Services Threats Prevented

- SQL Injections
- Capture and Replay Attacks
- DoS (resulting from a large load)
- Improper Error Handling
- Broken Access Control
- Large Payloads
- XPath Injections
- External Entity Attacks
- XML Bombs

Policy

- Disable DTD processing in XML parser

Test

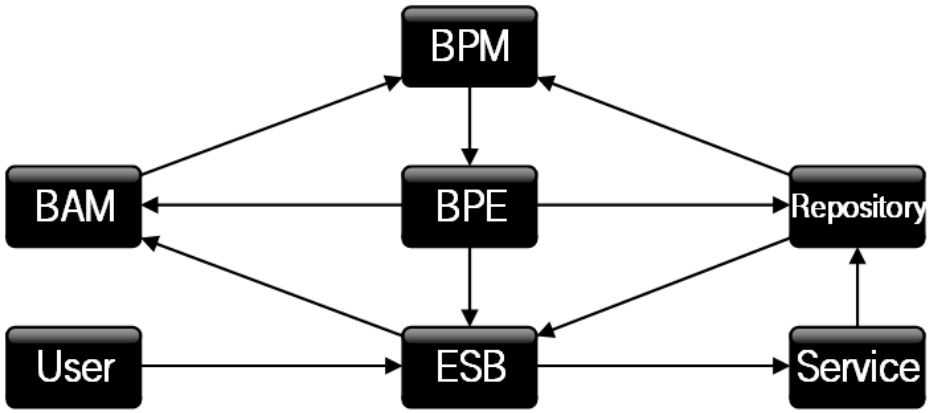
- Simulate attacks
- Regression test

T Systems

Protect and Attack

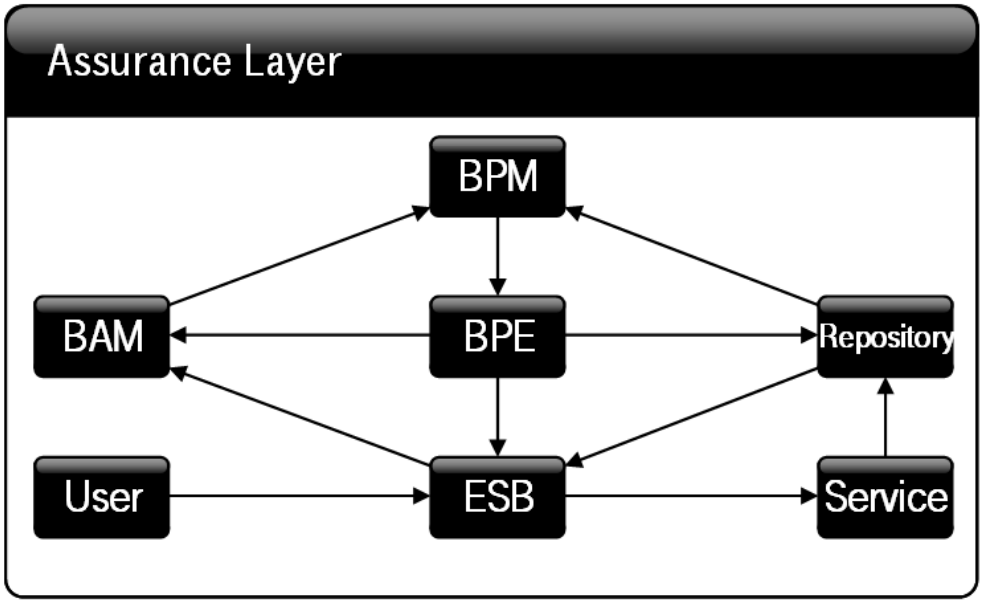
Protect and Attack

Protect



Protect and Attack

Protect

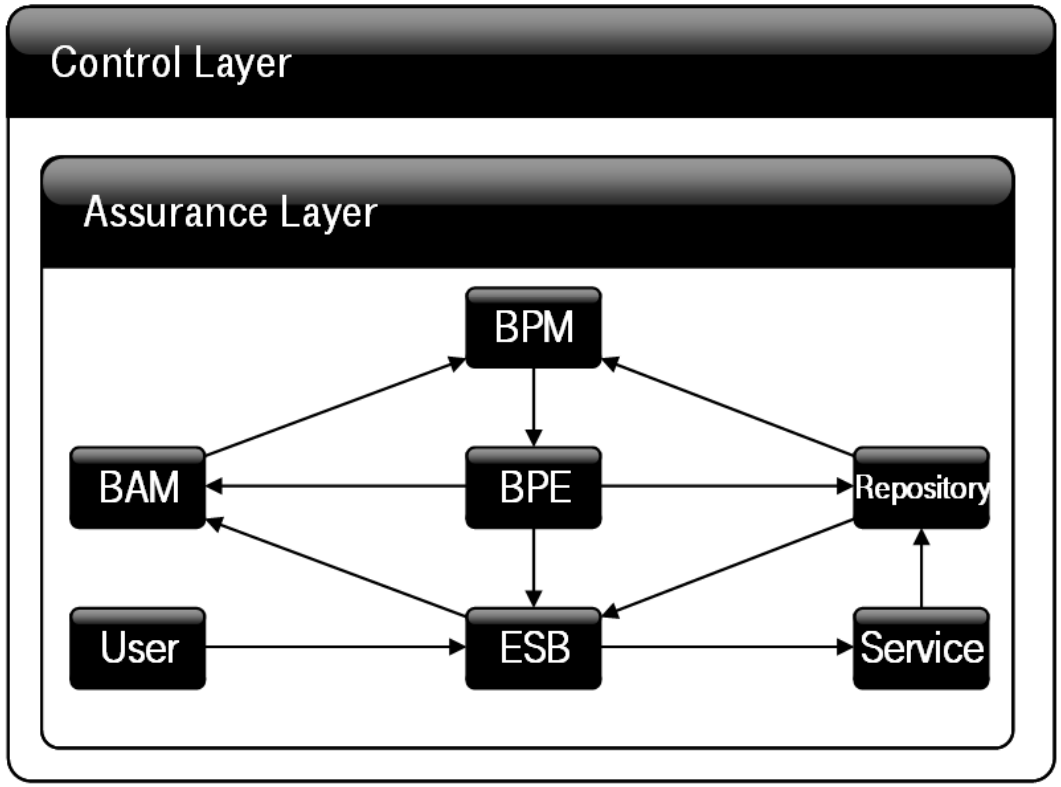


Organization

Can I comply with regulations?
Can I deliver audit reports?
Am I at risk?
Can I respond to security events?

Protect and Attack

Protect



Access Control

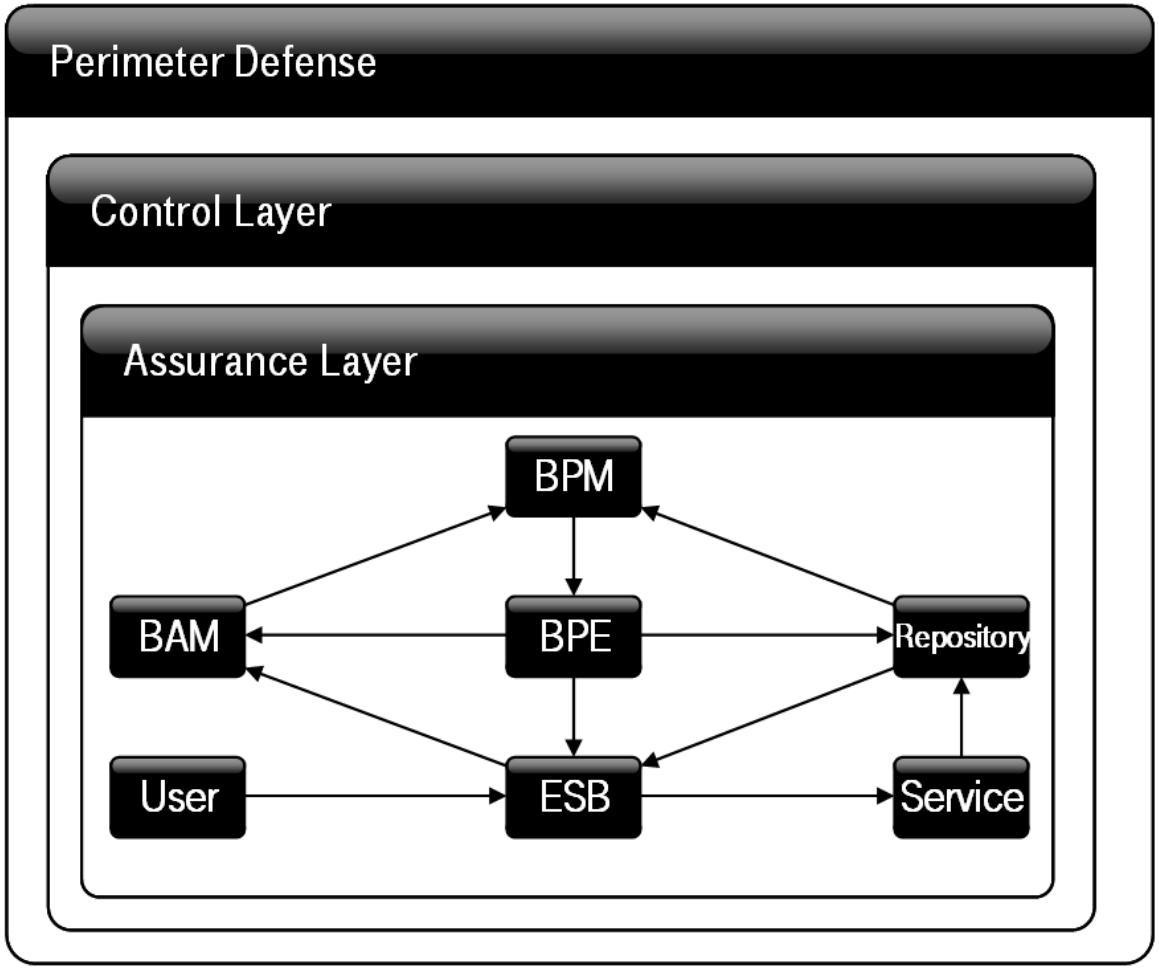
Which users can come in?
What can users see and do?
Are user preferences supported?
Can user privacy be protected?

Organization

Can I comply with regulations?
Can I deliver audit reports?
Am I at risk?
Can I respond to security events?

Protect and Attack

Protect



Keep out unwanted with

Firewalls
Anti-Virus
Intrusion Detection, etc.

Access Control

Which users can come in?
What can users see and do?
Are user preferences supported?
Can user privacy be protected?

Organization

Can I comply with regulations?
Can I deliver audit reports?
Am I at risk?
Can I respond to security events?

Protect and Attack

Attack

Internal Communication

- Approval by a manager
- Inform system operating
- Inform intrusion detection team
- Inform Firewall team

Protect and Attack

Attack

- Approval by a manager
- Inform system operating
- Inform intrusion detection team
- Inform Firewall team

- Hostname, IP-address, subnet, etc.
- Placement of penetration system (internal, external)

Protect and Attack

Attack

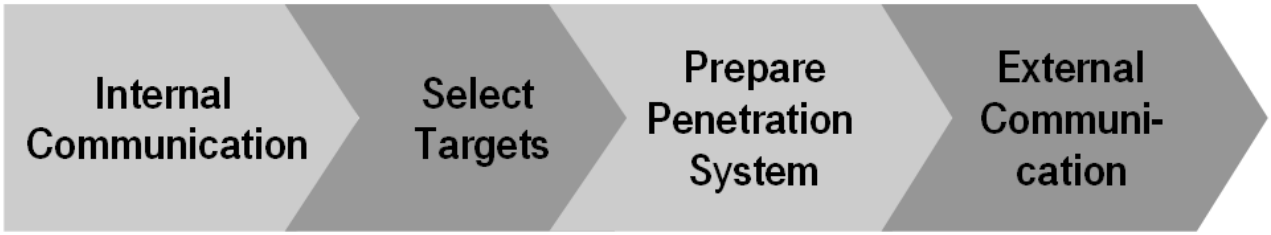
- Approval by a manager
- Inform system operating
- Inform intrusion detection team
- Inform Firewall team

- Hostname, IP-address, subnet, etc.
- Placement of penetration system (internal, external)

- Appropriate Tools

Protect and Attack

Attack



- Approval by a manager
- Inform system operating
- Inform intrusion detection team
- Inform Firewall team

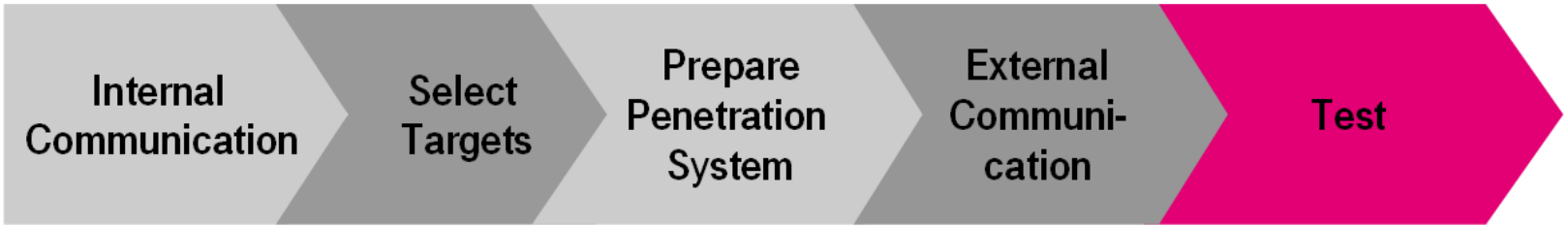
- Hostname, IP-address, subnet, etc.
- Placement of penetration system (internal, external)

- Appropriate Tools

- Inform your ISP Provider
- Inform Customers

Protect and Attack

Attack



- Approval by a manager
- Inform system operating
- Inform intrusion detection team
- Inform Firewall team

- Hostname, IP-address, subnet, etc.
- Placement of penetration system (internal, external)

- Appropriate Tools

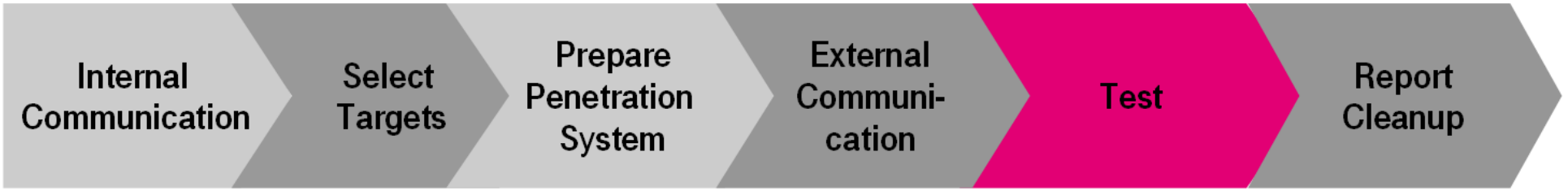
- Inform your ISP Provider
- Inform Customers

- Document what you did!

T Systems

Protect and Attack

Attack



- Approval by a manager
- Inform system operating
- Inform intrusion detection team
- Inform Firewall team

- Hostname, IP-address, subnet, etc.
- Placement of penetration system (internal, external)

- Appropriate Tools

- Inform your ISP Provider
- Inform Customers

- Document what you did!

- Document the results
- Fix possible problems
- Explain results to management

Thank you for your Attention!

Erleben, was verbindet.

Dominik Kopriva – 01.06.1983

- Diplom technische Informatik FH Regensburg 14.03.2010
- Diplomarbeit: Penetration von Webservices in einer SOA
- Expertise: SOA Security, Pentesting, Design / Implementierung inhomogener Computernetzwerke u.a.
- Ziel: IT-Security Consultant, Pentester, Netzwerkplanung/-management, SOA Architekt
- Kontaktinformationen:

  Email: DKopriva@gmx.net

  Tel.: 0173/6839203