KASPERSKY lab

An Effective Attack Method Based on Information Exposed by Search Engines

Antonios Gouglidis, University of Macedonia

"IT Security for the Next Generation"
European Cup, Prague
17-19 February, 2012

Kaspersky® Academy
IT Security
for the Next Generation
International Student Conference

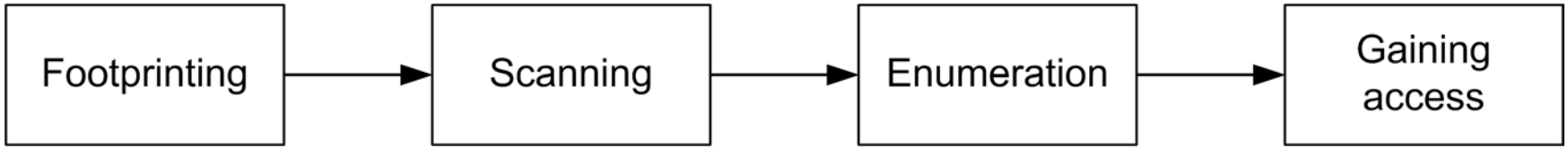
Motivation

- Extensive usage of Web 2.0 technologies
  - Mostly interested in WS provided by major search engines

- How WS can be used in a malicious way?

Anatomy of an Attack

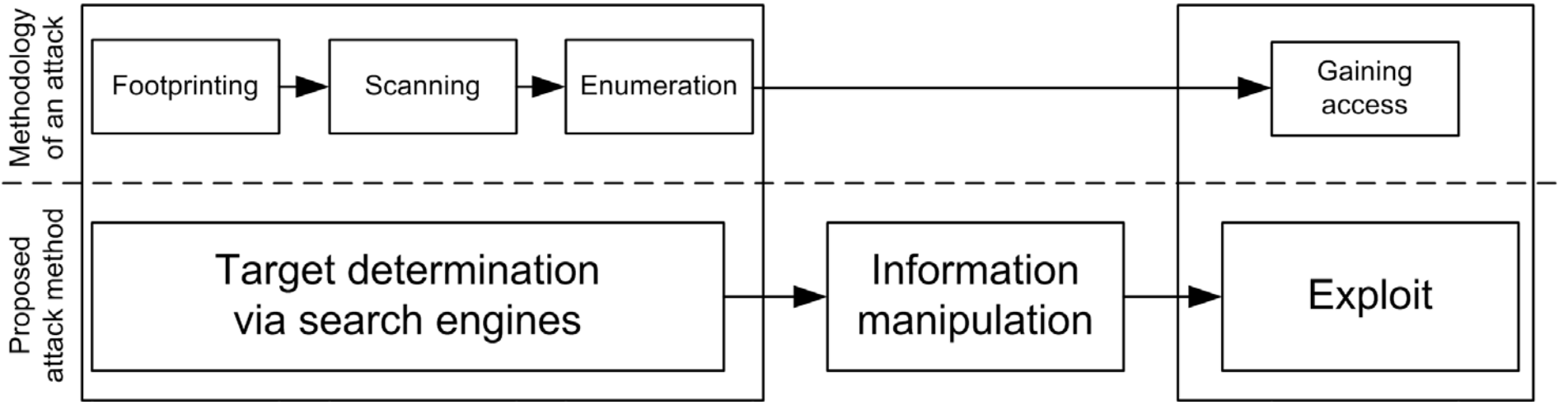
Initial Steps



	Footprinting	Scanning	Enumeration	Gaining access
Objective	Information gathering	Determination of reachable systems	Probe identified hosts and running services for known weaknesses	Attempt to access the target system
Technique	•Open source search •Whois •DNS zone transfer	•TCP/UDP port scan •OS detection •Ping sweep	•Identify applications •List file shares	•Buffer overflows •Password eavesdropping
Tools	•Search engines •UNIX/LINUX clients •nslookup	•nmap •fping	•Banner grabbing •showmount	•Bind, ISS •tcpdump

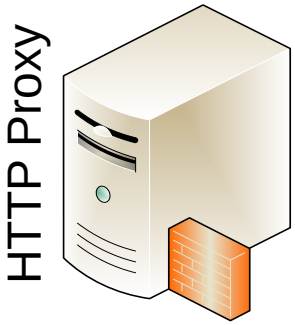
The Proposed Attack Method

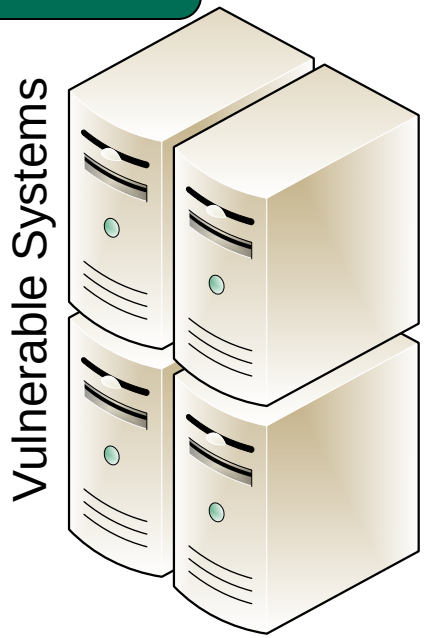
A 3-step Methodology



How to Deploy the Attack

The attack cannot be identified, until its deployment !!!





Create a query using advanced search operators

Return URLs

Create a valid HTTP Post Request

Deploy the exploit

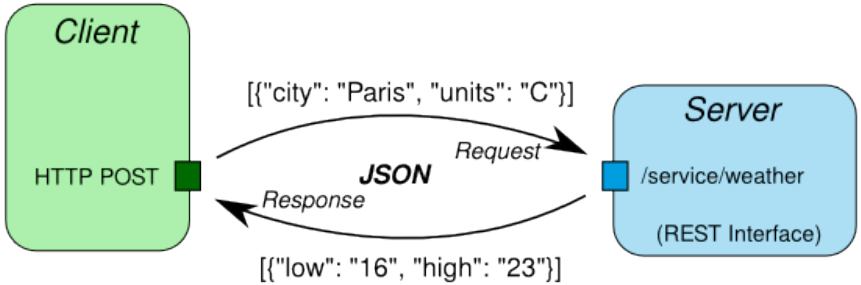
Implementation Prerequisites

- Register to get an APPID for either Google or Bing

- The proposed methodology utilizes:
  - The “Google Hacking” technique

  - Web 2.0 technologies
    - REST approach
    - JSON

JSON / REST / HTTP



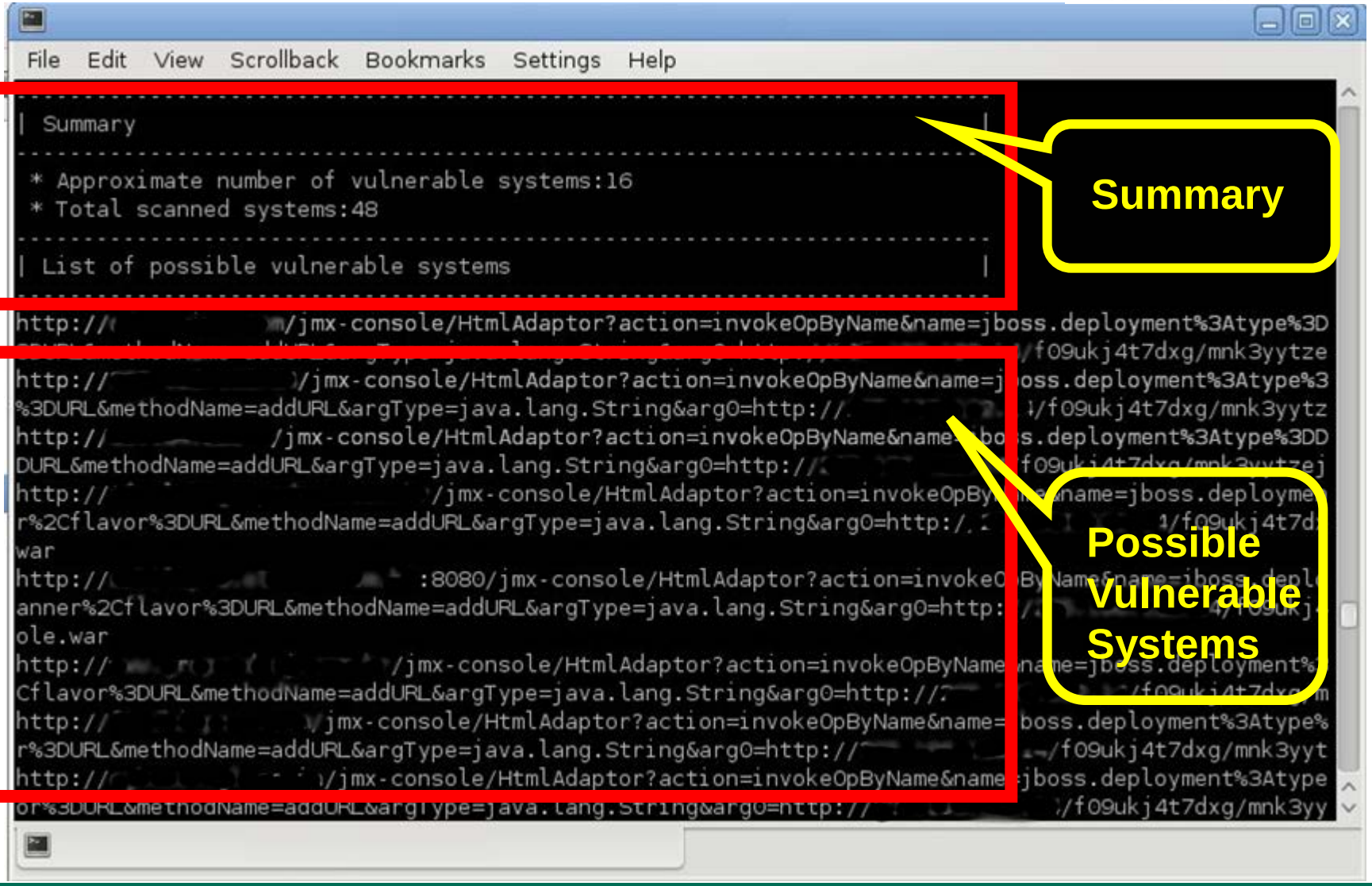
Implementation

JBossHacker.py

- Implemented in the Python scripting language
  - Approximately 50 lines of code
- Supported search engines
  - Google
  - Microsoft Bing
- What it can do?
  - Find servers having their JBoss JMX-Console open
  - Deploys an exploit
  - Gain command line access via a Web browser

Hands-on

JBossHacker.py - Results



Hands-on

JBoss Deployment Scanner

List of MBean attributes:

Name	Type	Access	Value	Description
URLComparator	java.lang.String	RW	org.jboss.deployment.Dep	MBean Attribute.
Filter	java.lang.String	RW	org.jboss.deployment.sca	MBean Attribute.
StateString	java.lang.String	R	Started	MBean Attribute.
State	int	R	3	MBean Attribute.
StopTimeOut	long	RW	60000	MBean Attribute.
ScanEnabled	boolean	RW	◉ True ○ False	MBean Attribute.
FilterInstance	org.jboss.net.protocol.URLLister$URLFilter	RW	org.jboss.deployment.sca	MBean Attribute.
URLList	java.util.List	RW	?12.170.156.148/cmd.war]	MBean Attribute.
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Name	java.lang.String	R	URLDeploymentScanner	MBean Attribute.
Deployer	javax.management.ObjectName	RW	jboss.system:service=Mai View MBean	MBean Attribute.
ScanPeriod	long	W		MBean Attribute.
URLs	java.lang.String	W		MBean Attribute.

Apply Changes

Hands-on

Gaining Command Line Access

How to defend yourself?

Existing Solutions

- Google Hack Yourself

- Rely on Policy and Legal Restrictions
- Google Diggity Project
  - Provides an Intrusion Detection System
    - Alert RSS Feeds
    - Alert RSS Monitoring Tools

Conclusions

The Proposed Attack Methodology

- What it can do?
  - Targets online Web Applications on the Internet
    - Not bounded to a single application
  - Deploy massive attacks, in an automated way
  - Undetectable until the time of deploying the exploit
  - High probability of a successful attack, if target satisfies *ALL* the criteria
- What it cannot do?
  - Discover new vulnerabilities
    - Prior knowledge of the vulnerability/exploit is required
  - No guarantees of a successful attack, if criterias are not met by the target

Thank You

Antonios Gouglidis, University of Macedonia

“IT Security for the Next Generation”
European Cup, Prague
17-19 February, 2012

KASPERSKY lab