



Application Penetration Testing: Concepts and Taxonomy

John Melton

Gail-Joon Ahn

University of North Carolina at Charlotte

College of Information Technology

Department of Software and Information
Systems



Background

- The number of bugs and critical vulnerabilities found in software is on the rise
- New software is created daily with security as an afterthought
- Attackers are always finding new ways to break into networks and applications



Objectives

- To identify techniques that can be used to perform penetration testing to help secure applications and networks
- To develop a taxonomy that allows professionals to easily recognize needed functionality when acquiring existing tools or when developing new tools



Two Types of Penetration Testing

- Zero-knowledge attack
 - Attacker has little or no knowledge of application
 - Accurately simulates most typical attacks
 - Usually performed by a trusted third-party
 - Correlates to “black-box testing”
- Full-knowledge attack
 - Attacker has much knowledge of application
 - Usually performed by internal employee
 - Correlates to “white-box testing”



Risks of Penetration Testing

- **Exposure to the public**
 - Third-party could leak any discovered vulnerabilities
 - Harms image and lowers the trust level of customers
- **False sense of security**
 - APT exposes vulnerabilities, but unlikely to find them all
 - Sense of “We tested it, so it must be secure” is wrong
- **Unskilled Testers**
 - Can crash networks, expose confidential information, and destroy information
- **Future attackers created**
 - The idea of teaching employees “how to hack”



Steps of Penetration Testing

- **Discovery (Also known as “fingerprinting”)**
 - Obtaining knowledge of an application
 - Includes application descriptions, usage, changelogs, etc.
- **Enumeration**
 - Extension of the Discovery step
 - Detailed study of the application as well as domain names, networks, and systems
- **Vulnerability Mapping**
 - Mapping all known vulnerabilities
- **Exploitation**
 - Attempting to break the application using vulnerabilities



Advantages of Penetration Testing

- Improve security by finding and plugging “holes” in an application
- Provides concrete evidence of the value of information security
- Helps to provide a return on investment value for information security (helps answer “Where is all the money going?”)



Metrics for Penetration Testing

- Information-gathering tools
 - Finding out useful information about applications, networks, or systems
 - Tools that can be used to perform the Discovery, Enumeration, and Vulnerability Mapping steps of penetration testing
- Attack tools
 - Actually perform an attack on an application
 - Tools that can be used to perform the Exploitation step of penetration testing



Metrics for Penetration Testing

- Information-gathering tools (detail)
 - Network exploration
 - Operating system fingerprinting
 - Determine available hosts/services/firewalls
 - Port scanning
 - Some network exploration tools map results graphically
 - Network sniffing
 - Examining network data (live or captured)
 - Searching network data on headers or payload
 - Functionality ranges from very basic to complex analysis logic
 - Remote security checks
 - Web application vulnerability scanning
 - Application fingerprinting
 - Analyze service to determine application and possibly version



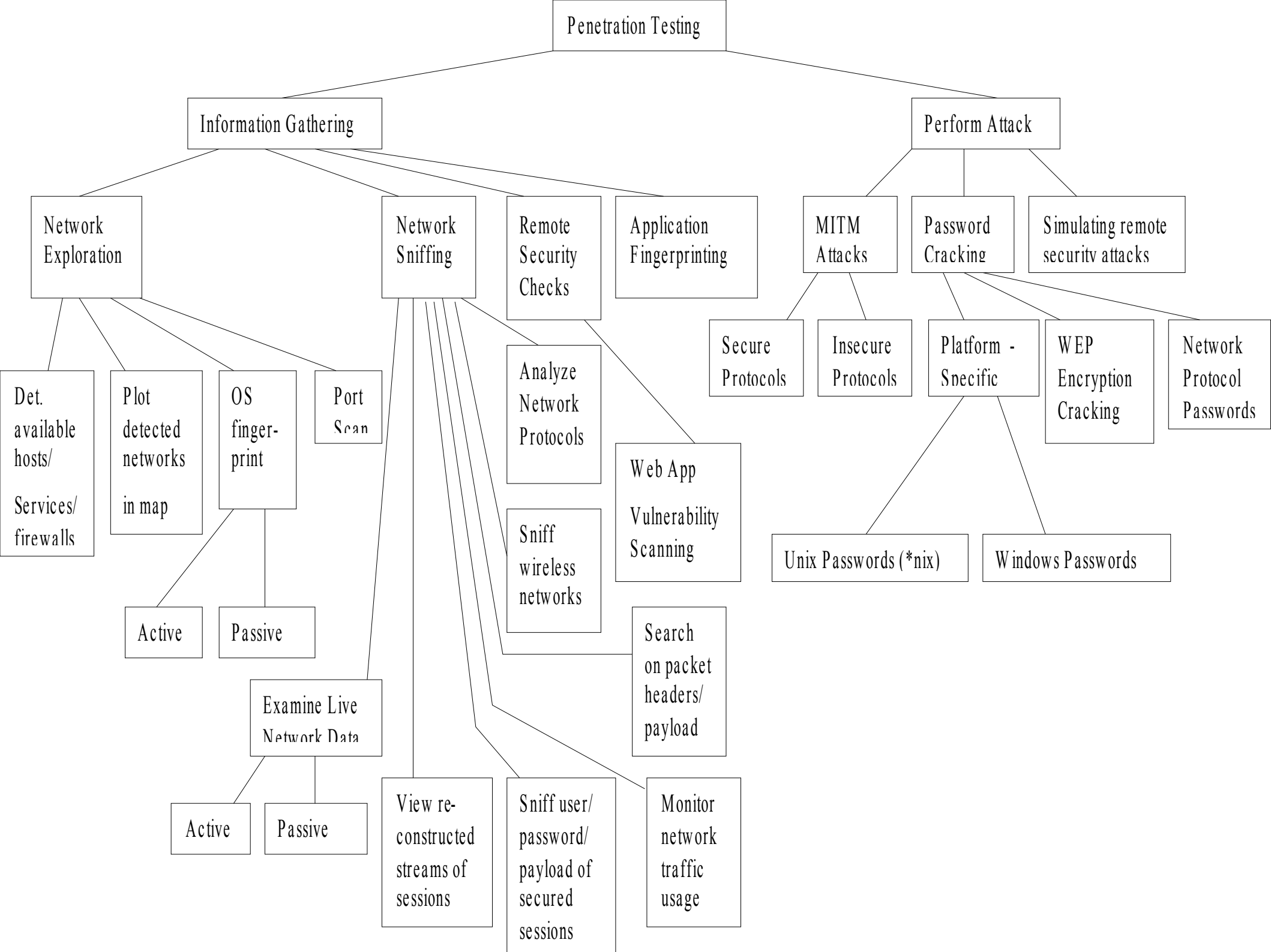
Metrics for Penetration Testing

- Attack tools (detail)
 - Man-in-the-middle attacks
 - Can be performed on encrypted or plain-text communications
 - Password-cracking attacks
 - Usually platform-specific
 - Many that work on network protocols as well as applications
 - Remote security attacks



Taxonomy

- Classifies several tools based on metrics
- Could be broken down further into platform-specific, environment-specific, etc.
- Should function as a guide to:
 - functionality to expect when attaining an existing tool
 - starting point of ideas for development of a new tool



Comparison

- Information-gathering tools vs. Attack tools
 - Information-gathering
 - Usually passive
 - Network aware
 - Often offer many capabilities (i.e. Able to sniff many network protocols)
 - Attack
 - Active
 - Often include functionality of info-gathering tools
 - Often depend on info-gathering tools output to perform
 - Usually focused on one particular attack
 - Overlap is common between groups of tools
 - Complement one another
 - Output of info-gathering tools often input to attack tools



Case Study 1

- Company A
 - Small technology firm – 25 employees
 - Starting to do web development, planning to host sites on internal network
 - Use taxonomy to find tools in these areas:
 - Information gathering
 - Network exploration
 - Remote security checks
 - Web application vulnerability scanning
 - Attack
 - Remote security attacks

Case Study 2

- Company B
 - Large financial institution
 - Struggling with wireless networking security issues
 - Use taxonomy to find tools in these areas:
 - Information gathering
 - Network exploration (specifically network plotting tools)
 - Network sniffing
 - Analyze network protocols
 - Sniff wireless networks
 - Attack
 - Password-cracking
 - WEP encryption cracking
 - Create new tool, place in information gathering under:
 - Network sniffing
 - Plot detected networks in a map
 - Plotting wireless networks



Case Studies – Lessons Learned

- Useful to see network from attacker's point of view
- Taxonomy is fairly broad and extensible
- Taxonomy can be applied in a very practical way, in several contexts

Conclusion

- Application penetration testing is a useful service that can greatly enhance the security stature of an organization
- The taxonomy provided can aid information security professionals in the penetration testing task in a practical way
- See full paper at :
http://www.coe.uncc.edu/~jtmelton/files/apt/apt_taxonomy.pdf

Open-Source Solutions

- Information-gathering tools
 - Network exploration
 - Determine available hosts/services – Nmap, Hping2, Ettercap, Firewalk, THC-Amap, Paketto Keiretsu
 - Plot detected networks in a map – Ettercap, Kismet, Firewalk, Paketto Keiretsu
 - OS fingerprinting – Nmap, Ettercap, Xprobe2
 - Port scanning – Nmap, Ettercap

Open-Source Solutions

- Information-gathering tools
 - Network sniffing
 - Examine network data (live or captured) – [Ethereal](#), [TCPDump](#), [NBTScan](#), [Ngrep](#), [Hunt](#)
 - View reconstructed streams of sessions - [Ethereal](#)
 - Sniff user/password/payload of secured sessions – [Dsniff](#), [Ettercap](#)
 - Analyze network protocols – [Ethereal](#), [Dsniff](#), [Ettercap](#)
 - Sniff wireless networks – [Ethereal](#), [Kismet](#), [Airsnort](#)
 - Search on packet headers/payload – [Ethereal](#), [TCPDump](#), [Ettercap](#), [Ngrep](#)
 - Monitor network traffic usage - [Ntop](#)

Open-Source Solutions

- Information-gathering tools
 - Remote security checks - [Nessus](#)
 - Web application vulnerability scanning – [WebScarab](#), [Nessus](#), [Whisker/Libwhisker](#), [Nikto](#), [SPIKE Proxy](#)
 - Application fingerprinting – [THC-Amap](#)

Open-Source Solutions

- Attack tools
 - Man-in-the-middle attacks – [Dsniff](#), [Ettercap](#), [Fragroute](#)
 - Password-cracking attacks – [John the Ripper](#)
 - WEP encryption cracking - [Airsnort](#)
 - Network protocol passwords - [THC-Hydra](#)
 - Remote security attacks – [WebScarab](#), [SPIKE Proxy](#)