Defense of the Assets

15 August 2015

Satria Ady Pradana

id

- Satria Ady Pradana
  - Teknik Informatika ITB 2010
  - PT LAPI Divusi – Ganesha Avionics division (ATC, radar & stuffs)
  - Interested in low level stuffs

satria.ady.33@gmail.com http://xathrya.web.id

agenda

- #1 Welcoming the Dark Side
  (introduction to cyber world and how to survive)
- #2 Know Your (Potential) Enemy!
  (talk about risk, threats, and assets)
- #3 Hardening and Defending <Blue Team>
- #4 Do Haxor Way <Red Team>

There are some demos and handons.

Provided Material

- Virtual Image (OVA file, compressed)
  - Arch Linux 64-bit
  - LAMP Stack
  - Example Codes of 4th session
  - This presentation
- Vulnerable VM

Distributed for free, ask official.

satria.ady.33@gmail.com http://xathrya.web.id

#1 Welcoming the Dark Side

World is big, but cyber world is BIGGER!

- Boundless, no spatial border
- Identity?
- Trust?
- Anything can be there and anything can be done. Did I said anything?

Bad people exists and you should worry!

You can be anyone and you can be no one

satria.ady.33@gmail.com http://xathrya.web.id

So Many Colors

- White (Hat)
- Gray (Hat)
- Black (Hat)
- Red (Team)
- Blue (Team)
- etc

Be Defender

- Know why you do this.
- Know how attacker attacks.
- Know how to defend yourself, your assets, etc.
- Know why it can be like this.
  (If you are screwed, at least you know why)

#TeamDefender

The Key People

- Security Engineer Team
  (who design the system)
- Incident Response Team
  (in case of breach, call them immediately)
- Digital Forensic Team
  (they are “detectives”)
- Auditor
  (checking your system for holes)

#TeamDefender

Be Attacker

- Know why and how you do this.
- Know how system works.
- Know why it can be like this.
- Know how defenders defend their selves and what will they do.

Sometimes attacker have a step ahead.

#TeamAttacker

Red or Blue? Choose Wisely

- Be Defender?
- Be Attacker?

Why not both?

There are skills you need to acquire.

Common Skills Required

- Programming
- Operating System and Services
- Network and Communication
- Security Concept
- Cryptography
- Common Architecture Design
- Reverse Engineering
- Digital Forensics

Is it all?

- Creative Thinking
- Problem Solving
- Persistence
- Common Mistake and Best Practice Knowledge

satria.ady.33@gmail.com http://xathrya.web.id

#2 Know Your (Potential) Enemy!

Wait, we should discuss about these first:

- CIA Triads
- Assets
- Threats

In short: Security Concept.

CIA Triads

- Confidentiality
  - Information should be known only to right people.
- Integrity
  - One can determine whether the data received is original, unmodified and unaltered on the way.
- Availability
  - System or information is guaranteed to be available when needed.

Assets

- Any component in business logic.
- Data, Information, Devices, Schemas, Important Letters, etc.
- Mostly it’s about data.
- May or may not related to actual or physical object.

Threats

- Possibility of breach.
- Breach = loss, reduce value of assets.
  - Money
  - Power
  - Business opportunity
  - Reputation
  - Etc.

- If you have valuable good in your possession, expect bad people want it.
- Your protection is as good as your weakest link.
- Threats can’t be removed completely, it can only be reduced or minimized.

So who's your enemy?

- Internal
- External

Trust no one, nor zero.

Popular Term

Before we start understanding the concept, we must speak the language.

- Bug
  error, flaw, failure, or fault which produce an incorrect or unexpected result, or unintended behavior.
- Vulnerability
  flaw in system's security that can lead attacker to utilizing the system in a manner other than the designer intended.
- Exploit
  tool, set of instructions, or code which take advantage of vulnerability.

Not all bug lead to vulnerability

satria.ady.33@gmail.com http://xathrya.web.id

#3 Hardening and Defending

#TeamDefender

- Mission : Defending Assets
- Various Level :
  From abstract to the concrete level.

Architectural View

- The design of whole system
- Might involve Policy
- The most critical part
  (design error might result in catastrophe)

Security Design Principles

- Least Privilege
  (accomplish task with the least privilege you can)
- Fail-Safe Defaults
  (if the system fail, it should has a mechanism to assured system not break)
- Open Design
  (use design which is widely approved as good design)
- Privilege Separation
  (don’t mix privilege)
- Defense in Depth
  (multi layered defense, never delegate it to single defense only)

Component View

We talk about Hardening

- Process of enhancing server security through a variety of means which results in a much more secure server operating environment.
- When server is put online, hardening is a must.

We use ArchLinux as instance.

Hardening Steps

- Reconnaissance
- Vulnerability Mapping
- Planning
- Execution
- Evaluating

Stage 1: Reconnaissance

Gathering information, search for valuable information related to our task. Anything which can help our work.

- Assets in the server
  (what we protect)
- Network topology
  (how our server can be accessed)
- Server spesifics
  (OS, kernel, important drivers, existing services, etc)
- Users
  (who had privilege over resources, who use the system)
- Etc.

Stage 2: Vulnerability Mapping

Mapping threats and potential breach to information found.

- Will this service susceptible to this threat?
- What can affect this service?
- What vulnerability are found for my current service version (and also past version).

In the end you should know what might disrupt your system. Even if you don't know the specific, imagine what can harm your server.

Stage 3: Planning

Plan all things we will do, define some goals, and how we can evaluate our it.

- What you want to achieve?
- What is your priority?
- How to evaluate goal?

Stage 4: Execution

Time to do the hard work.

- Patch all known vulnerability.
- Remove unused service.
- Recheck configuration and evaluate all rules given.
- Gives extra protection if necessary.
- Follow some best practice.
- Write down all your work.

Stage 5: Evaluating

Decide whether you have enough, using your parameters.

Often, it is verified by penetration testing mean.

Example and Exercise

Physical Security

- Configure BIOS to disable booting form external media (CD/DVD, floppy drive, flash drive, etc).
- Encrypt partition (if necessary).
- Give root a password
  prevent single mode access with no authentication.

System Updates

[1] Keep system updated!

```
pacman -Syy
pacman -Su
```

Roughly, equivalent to these commands on Debian/Ubuntu

```
apt-get update
apt-get upgrade
```

ps: ArchLinux is rolling release

Users

[1] Use shadow user with sudo instead of root account.

Create user (ex: xathrya)
sudo useradd –d /home/xathrya –s /bin/bash –m xathrya
Give sudo access
sudo usermod –a –G sudo xathrya
Set password
sudo passwd xathrya

Remember to use proper password.

[2] Disable root account so outsider can’t make use of it.

Lock it.

sudo passwd –l root

If you want to unlock.

sudo passwd –u root

[3] Disable shell for active account which is not actual user (irc, eggdrop, bnc, ptlink, guardservices, ftp, etc).

See active accounts

```
cat /etc/passwd | egrep –v '\/false|\/nologin|\/shutdown|\/halt' | cut –d':' –f 1,7
```

Disable account

```
usermod –s /usr/sbin/nologin username
```

Connection & Access

[1] Secure console

Limit where you can login by restricting which terminal you want to use. Allow only one terminal.

Edit /etc/securetty and comment all other terminals using # sign.

Make root the only one who can modify it.

sudo chown root:root /etc/securetty

sudo chmod 0600 /etc/securetty

[2] Make SSH listening on alternate port

Edit /etc/ssh/sshd_config

Search for Port 22 and change it to arbitrary port.

[3] Use PAM module for SSH

[4] Port Knocking

Only open port when you have “knocked” some specific ports.

[5] Slow Response

Response time exponentially for breach attempt.

Secure Shared Memory

Shared memory can be used in attack against a running service.

Modify /etc/fstab and add following line:

```
tmpfs   /run/shm   tmpfs
defaults,noexec,nosuid   0   0
```

Securing LAMP Stack

LAMP = Linux + Apache + MySQL + PHP

- Apache Hardening
- MySQL Hardening

Apache Hardening

- Latest version, please.
- Hide version number
  (or you can even fake it for pr0fit)
  edit httpd.conf and modify

```
ServerSignature Off
ServerTokens Prod
```

- Run Apache under its own user account

- Ensure other than web root is not served.
- Turn off directory browsing
- Turn off server side includes
- Turn off CGI execution
- No follow symbolic links
- Run mod_security module and disable unnecessary modules

- Only root can read apache’s config and binaries
- Lower timeout value
- Limiting large requests
- Limiting size of XML body
- Run Apache in CHROOT environment (tricky, use mod_security)

- Enable logging
- Change root directory

MySQL Hardening

- Run MySQL under its own user account
- Bind to proper address, ex: only bind to localhost
- Disable LOCAL INFILE
- Change ROOT username and password
- Remove “test” database
- Remove Anonymous and obsolete accounts

Question? #1

satria.ady.33@gmail.com http://xathrya.web.id

TAKE A BREAK

satria.ady.33@gmail.com http://xathrya.web.id

#4 Do Haxor Way

#TeamAttacker

- Hacking Steps
- Diving to the Heart of Machine
  - Memory model
  - Buffer Overflow
  - Exploit
- Introduction to metasploit
- Exploiting vulnerable VM

Hacking Steps

We call it penetration testing.

- Reconnaissance & Analysis
- Vulnerability Mapping
- Gaining Access
- Privilege Escalation
- Maintaining Access
- Covering Tracks

Stage 1: Reconnaissance

Gathering information, search for valuable information related to our target. Analyze the target from publicly available sources.

- Publicly exposed machine
  (which one we available to us)
- Open port
  (available door to us in)
- Network
  (relation of other systems)
- Server spesifics
  (OS, kernel, important drivers, existing services, etc)
- Users
  (who might had privilege over resources, ex: HR manager)
- Etc.

Stage 2: Vulnerability Mapping

Mapping threats and potential breach to information found.

- Based on the system we found, what threat available?
- How can we conduct attack?
- Make priority from the list, decide which one give greater chance of success.

Simulate scenarios to break in before we get to the next stage.

Stage 3: Gaining Access

The actual penetrating phase. Our purpose is to break in, using the vulnerabilities found in previous steps.

Stage 4: Privilege Escalation

When we break in, we might not have enough privilege to take over. Therefore, we need to exploit other thing to take higher privilege.

Stage 5: Covering Tracks

Don’t let any trace left.

- Delete logs
- Fabricate logs
  (smarter yet trickier way)

Create fake evidence.

- Memory and Pool
- File

Memory Model

- Program = data + instruction
- Process = instance of program running in memory
- Heap and Stack area are dynamic
- Memory spaces are marked by flag: writeable, executable.

Stack Layout

Buffer Overflow

- Fill buffer over the amount it can hold.
- No proper bound checking.
  What if we overwrite the Return Address?

Exploit (demo)

Introduction to Metasploit (demo)

Exploiting vulnerable VM (demo)

Question? #2

satria.ady.33@gmail.com http://xathrya.web.id

Beyond!

Let’s talk about what can we do in future…

- Distributed and parallel system
- Services architecture

Thanks!

satria.ady.33@gmail.com http://xathrya.web.id