How to Response Against Web Security Incident

Digit Oktavianto
digit dot oktavianto at gmail dot com
http://digitoktavianto.web.id

BSSN – 11th August 2018

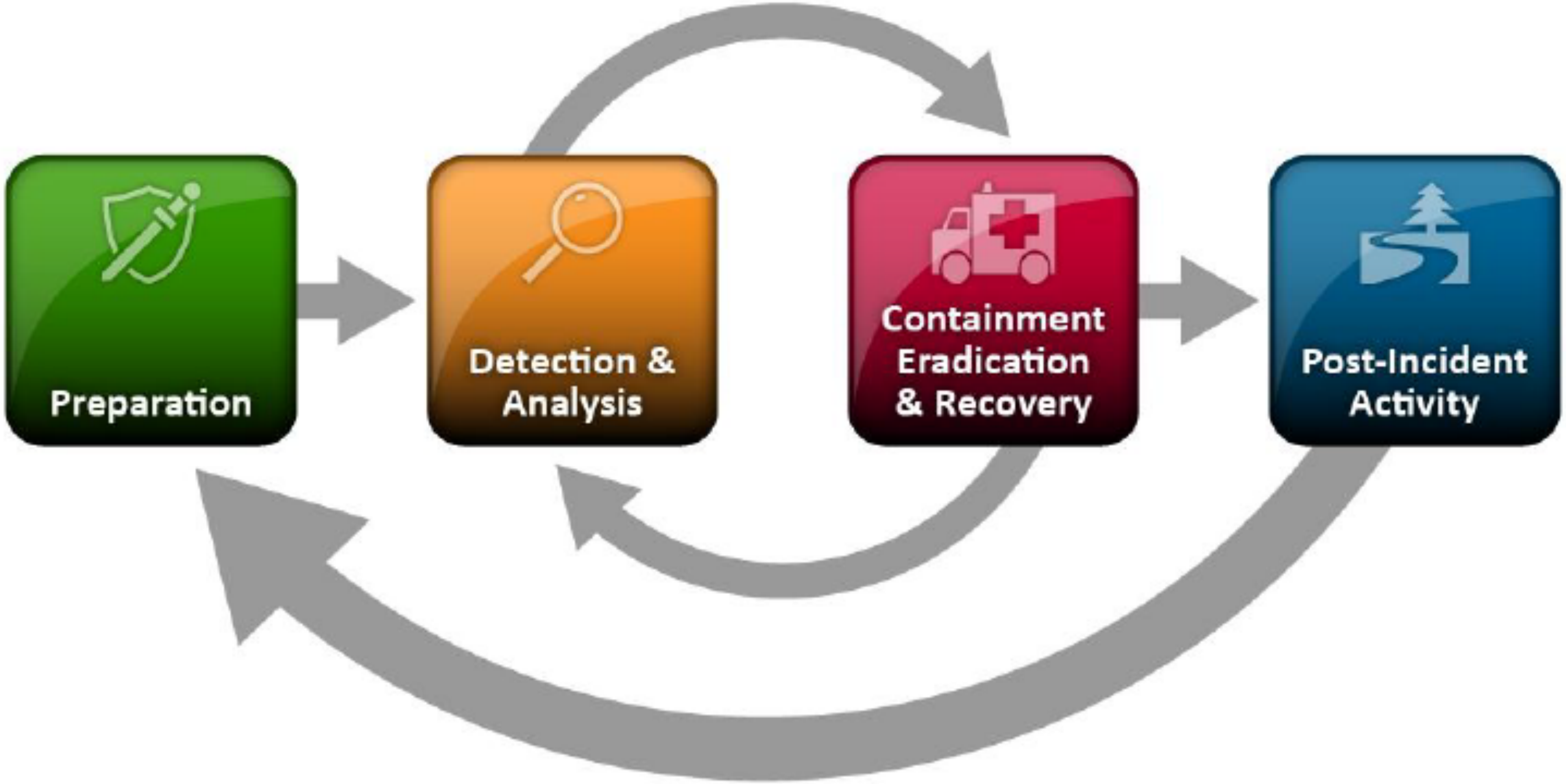
Agenda

- Incident Response Life Cycle Recap
- Incident Response Web Hacking PlayBook
- Incident Response Step for Web Hacking Security Incident
- What to Do After IR Step is Done.

NIST SP 800-61rev2 Incident Response

- An ***event*** is any observable occurrence in a system or network. Events include a user connecting to a file share, a server receiving a request for a web page, a user sending email, and a firewall blocking a connection attempt.
- ***Adverse events*** are events with a negative consequence, such as system crashes, packet floods, unauthorized use of system privileges, unauthorized access to sensitive data, etc.
- ***computer security incident*** is a violation or imminent threat of violation1 of computer security policies, acceptable use policies, or standard security practices.

Incident Response Life Cycle (NIST)



Preparation

Detection & Analysis

Containment Eradication & Recovery

Post-Incident Activity

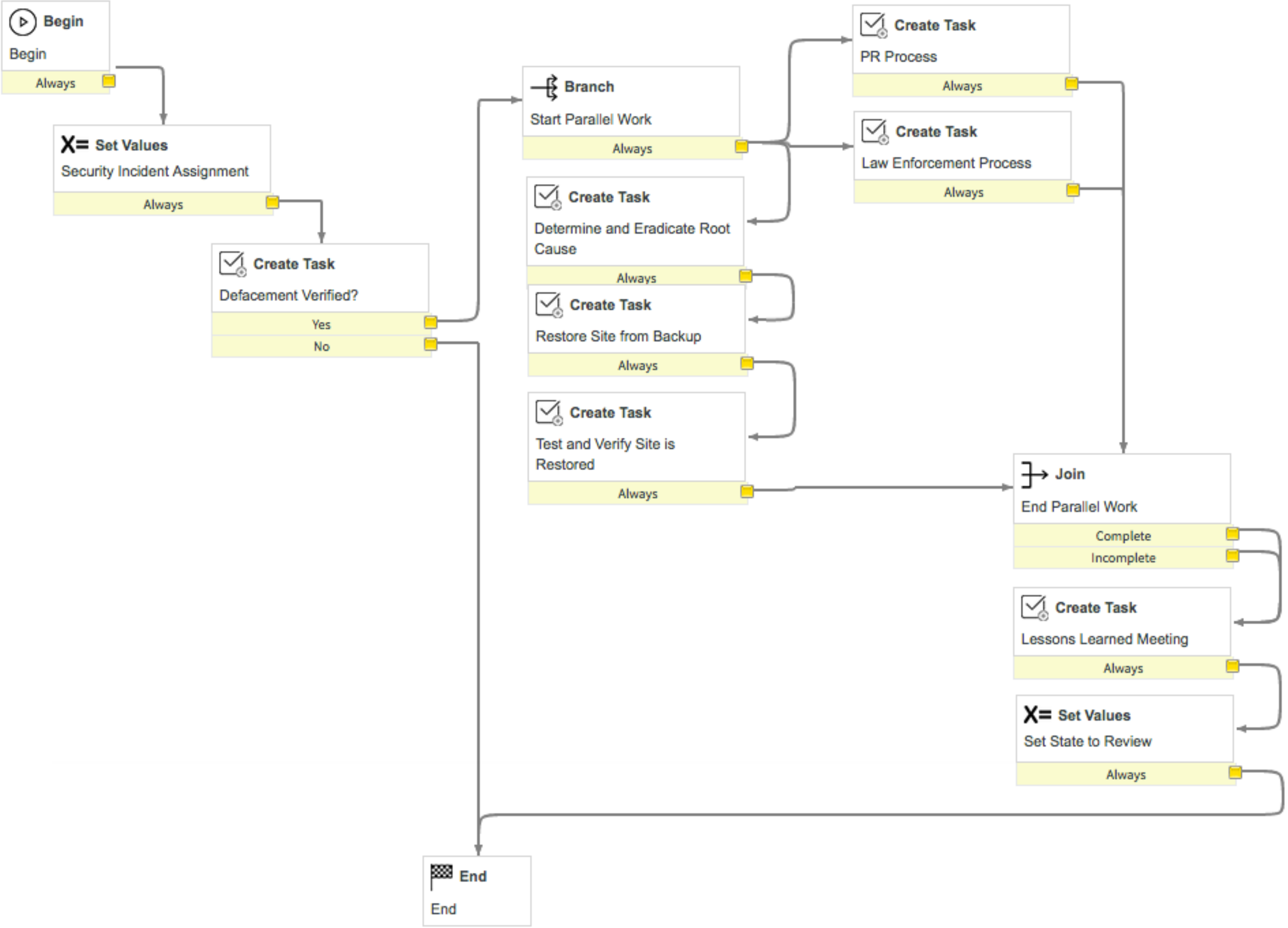
IR Life Cycle Recap

- **Preparation**: get ready to handle the incident
- **Identification**: detect the incident
- **Containment**: limit the impact of the incident
- **Eradication**: remove the threat
- **Recovery**: recover to a normal stage
- **Lesson Learned**: draw up and improve the process

Tips for Building Effective Incident Handling Plan (Cont'd)

- Improve vulnerability management Program
- Learn from past incidents and breaches
- Improve incident handling workflow process
- Building centralized monitoring system to protect the infrastructure

Web Hacking IR PlayBook



Begin
Begin
Always

X= Set Values
Security Incident Assignment
Always

Create Task
Defacement Verified?
Yes
No

Branch
Start Parallel Work
Always

Create Task
Determine and Eradicate Root Cause
Always

Create Task
Restore Site from Backup
Always

Create Task
Test and Verify Site is Restored
Always

Create Task
PR Process
Always

Create Task
Law Enforcement Process
Always

Join
End Parallel Work
Complete
Incomplete

Create Task
Lessons Learned Meeting
Always

X= Set Values
Set State to Review
Always

End
End

Root Cause Web Security Incident

Most Common Root Cause Problem for Web Hacking Incident :

- Vulnerability in Web Apps Itself
- Vulnerability in 3rd Party Component Used by Developer (Plugin, AddOn Module, etc)
- Unpatched Operating System
- Vulnerability in Services of OS (Web Server Vuln, DB Server Vuln, etc)

IR Step for Web Hacking

- **Preparation**: Prepare before Incident Happen and Ready to Handle Web Security Incident
- **Identification**: detection of Web Hacking Security Incident
- **Containment**: Limit the impact of the Web Hacking Security Incident
- **Eradication**: Removing the Root Cause of Web Hacking Security Incident
- **Recovery**: recover to a normal stage
- **Lesson Learned**: draw up and improve the process

IR Step : Preparation

Objective: Establish contacts, define procedures, and gather information to save time during an attack.

- Have up-to-date schemes describing your applicative **components related to the web server**.
- Build a **backup website up and ready**, on which you can publish content.
- Define a procedure to redirect every visitor to this backup website for Disaster Recovery Plan
- Deploy **monitoring tools to quickly detect any abnormal behaviour** on your critical websites.

IR Step : Preparation (Cont'd..)

- **Export the web server's log files to an external server (Log Management Server / SIEM)**. Make sure **clocks are synchronized between each server**.
- Reference external contents (static or dynamic) and create a list for each of them.
- Reference **contact points of your hosting provider**.
- Be sure your hosting provider enforces policies to log all events.
- Make sure you have an **up-to-date network map**.

IR Step : Preparation (Cont'd..)

Sample Technical Activity for Preparation Phase :

1. Forward Syslog from OS and Access Log from Web Apps Log to Log Management Server / SIEM.
   - Tools : Rsyslog Client and Rsyslog Server ; Filebeat and ELK ; Nxlog ; OSSEC / Wazuh for Host IDS Log.
2. Sync Clock Between Server to NTP Server
   - Tools : Ntpd Service (Linux) ;
3. Backup Regularly
   - Tools : Bacula ; Amanda
4. Monitoring Tools at Endpoint Server
   - Tools : OSSEC (HIDS and FIM) ; Wazuh (HIDS and FIM) ; Sysmon (Windows) ; Samhain (File Integrity Monitoring)

IR Step : Identification

Objective: Detect the incident, determine its scope, and involve the appropriate parties.

Usual channels of detection are:

- **Webpage monitoring**: The content of a web page has been altered. The new content is either very discreet (an "iframe" injection for example) or obvious ("*Hacked by XXX Crew*")
- User: **users call or notification** from employees about problems they noticed while browsing the website.
- Security checks with tools such as **Google SafeBrowsing**

IR Step : Identification (Cont'd..)

Verify the defacement and detect its origin:

- Check files with static content (in particular, check the modification dates, hash signature).
- Check mashup content providers.
- Check link presents in the web page (src, meta, css, script, …).
- Check log files.
- Scan the databases for malicious content.

IR Step : Identification (Cont'd..)

Sample Technical Activity for Identification Phase :

1. Check history command in terminal :
   - # history
2. Check All Logs :
   - **OS Log (/var/log/messages ; /var/log/dmesg)**
   - **Authentication Log (/var/log/auth.log** ; **/var/log/lastlog ; /var/log/btmp ; last –f /var/log/wtmp** or last –f **/var/log/utmp ; /var/log/secure ;**)
   - **Web Access Log (/var/log/apache2/access.log ; /var/log/apache2/error.log)**
3. Check Network Connection :
   - Netstat Command : **netstat -plant**
4. Check Process List :
   - Ps command : **ps -aux**

IR Step : Identification (Cont'd..)

Sample Technical Activity for Identification Phase :

5. Check Open Files :
   - Lsof command : **lsof -p (pid) ; lsof -i** (Look for unusual port listen)
6. Check User Account Registered for Susspicious User:
   - Look at /etc/passwd : **cat /etc/passwd**
7. Check Scheduler Task:
   - Crontab File : **cat /etc/crontab ; ls /etc/cron.* ; ls /var/at/jobs**

IR Step : Containment

Objective: Mitigate the attack's effects on the targeted environment.

- **Backup all data** stored on the web server for forensic purposes and evidence collecting. The best practice here if applicable is to make a complete bit-by-bit copy of the hard-disk containing the web server. This will be helpful to recover deleted files.
- **Check your network architecture map. Verify that the vulnerability exploited by the attacker is not located somewhere else :**
  - Check the system on which the web server is running,
  - Check other services running on that machine,
  - Check the connections to other systems, which might be compromised.

  If the source of the attack is another system on the network, disconnect it if possible physically and investigate on it.

IR Step : Containment (Cont'd..)

Try to find evidences of every action of the attacker:

- **Find out how the attacker got into the system in the first place and fix it :**
  - Web component vulnerability allowing write access: fix the vulnerability by applying the fix.
  - Open public folder: fix the bug.
  - SQL weakness allowing injection: correct the code.
  - Mashup components: cut mashup feed.
  - Administrative modification by physical access: modify the access rights.
- **If required (complex issue and very important web server), deploy a temporary web server**, up to date with its applications. It should offer the same content than the compromised web server or at least show another legitimate content such as ***"Temporary unavailable".*** The best is to display a temporary static content, containing only HTML code. This prevents another infection in case the attacker has used vulnerability in the legitimate PHP/ASP/CGI/PL/ etc. code.

IR Step : Containment (Cont’d..)

Sample Technical Activity for Containment Phase :

1. Move / Change Hacked Page / Defaced Page to Temporary Unavailable Page (Change in A Record / CNAME in DNS Configuration)
2. Redirect Hacked Website to Temporary Page / Another Server
3. Disconnect Hacked Web App Server from Network

IR Step : Eradication / Remediation

Objective: Take actions to remove the threat and avoid future defacements.

- **Remove all altered content** and replace it with the legitimate content.
- Fixing the finding of vulnerability
- Restored content from earlier backup. Make sure this content is already free from vulnerabilities (if vuln sources is from web apps itself).

IR Step : Eradication (Cont'd..)

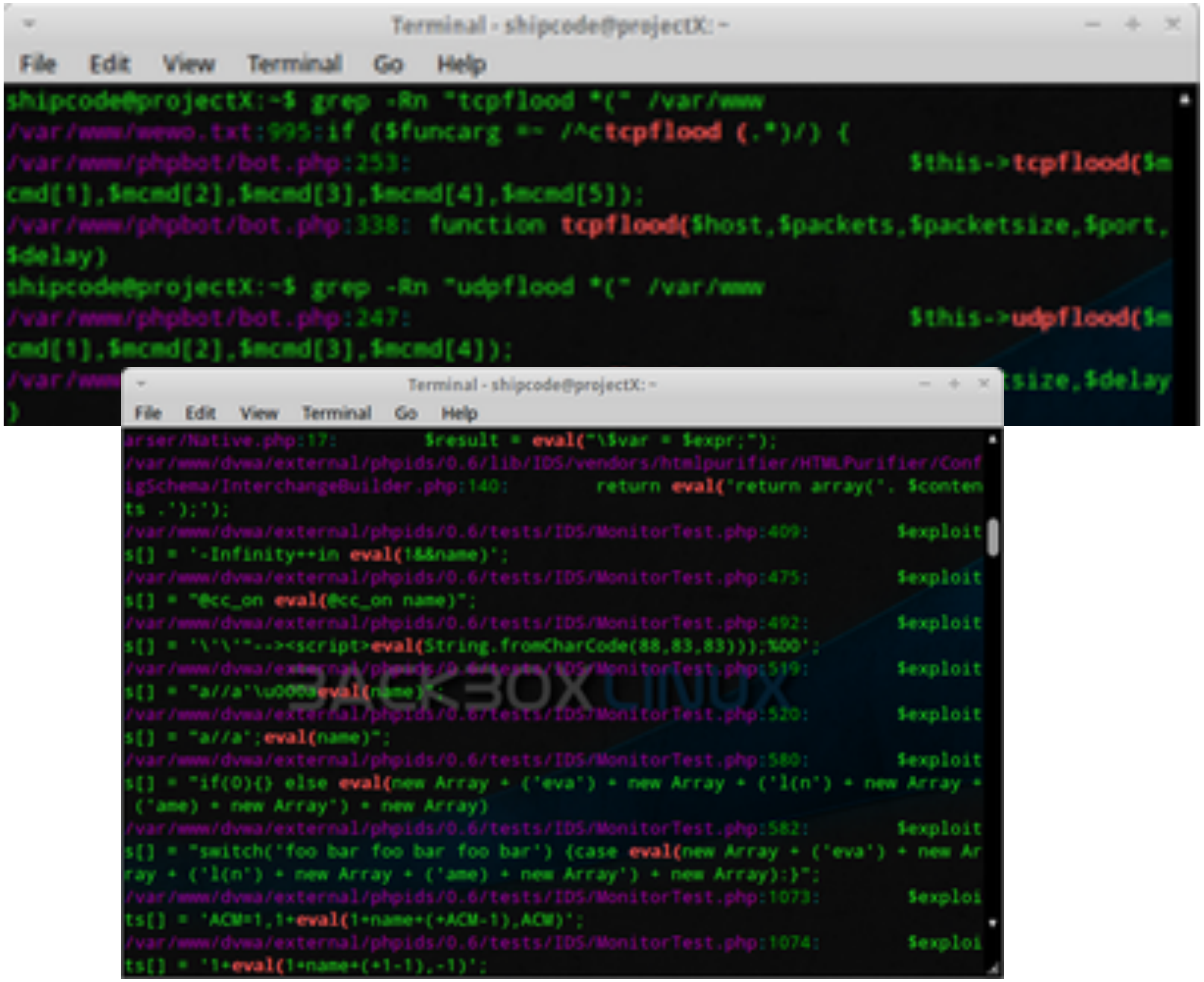
Sample Technical Activity for Eradication Phase :

1. Remove Hacked Page and Change the Normal Page
2. Search / Hunting the Backdoor (**Details in Next Page**) and Remove the Backdoor
3. Look for Suspicious process and remove the OS Backdoor / Rootkit :
   - Chkrootkit : http://www.chkrootkit.org/
   - Rkhunter : http://rkhunter.sourceforge.net/
   - Linux Malware Detect : https://github.com/rfxn/linux-malware-detect
   - MalDet : https://github.com/dkhuuthe/MalDet
   - ClamAV : https://www.clamav.net/
   - MalScan : https://github.com/mtingers/malscan
   - NeoPi : https://github.com/Neohapsis/NeoPI

Manual Finding Shell Backdoor

- **Checking PHP Backdoor / Web Shell / Backdoor Shell in Advanced :**
- grep -Rn “shell_exec *(” /var/www
- grep -Rn “base64_decode *(” /var/www
- grep -Rn “phpinfo *(” /var/www
- grep -Rn “system *(” /var/www
- grep -Rn “php_uname *(” /var/www
- grep -Rn “chmod *(” /var/www
- grep -Rn “fopen *(” /var/www
- grep -Rn “fclose *(” /var/www
- grep -Rn “readfile *(” /var/www
- grep -Rn “edoced_46esab *(” /var/www
- grep -Rn “eval *(” /var/www
- grep -Rn “passthru *(” /var/www

Manual Finding Shell Backdoor



Terminal - shipcode@projectX: ~
File Edit View Terminal Go Help

```
shipcode@projectX:~$ grep -Rn "tcpflood *(" /var/www
/var/www/wewo.txt:995:if ($funcarg =~ /^ctcpflood (.*)/) {
/var/www/phpbot/bot.php:253:                                        $this->tcpflood($m
cmd[1],$mcmd[2],$mcmd[3],$mcmd[4],$mcmd[5]);
/var/www/phpbot/bot.php:338: function tcpflood($host,$packets,$packetsize,$port,
$delay)
shipcode@projectX:~$ grep -Rn "udpflood *(" /var/www
/var/www/phpbot/bot.php:247:                                        $this->udpflood($m
cmd[1],$mcmd[2],$mcmd[3],$mcmd[4]);
/var/www                                                              size,$delay
)
```

Terminal - shipcode@projectX: ~
File Edit View Terminal Go Help

```
arser/Native.php:17:        $result = eval("\$var = $expr;");
/var/www/dvwa/external/phpids/0.6/lib/IDS/vendors/htmlpurifier/HTMLPurifier/Conf
igSchema/InterchangeBuilder.php:140:          return eval('return array('. $conten
ts .');');
/var/www/dvwa/external/phpids/0.6/tests/IDS/MonitorTest.php:409:          $exploit
s[] = '-Infinity++in eval(1&&name)';
/var/www/dvwa/external/phpids/0.6/tests/IDS/MonitorTest.php:475:          $exploit
s[] = "@cc_on eval(@cc_on name)";
/var/www/dvwa/external/phpids/0.6/tests/IDS/MonitorTest.php:492:          $exploit
s[] = '\"\'"--><script>eval(String.fromCharCode(88,83,83)));%00';
/var/www/dvwa/external/phpids/0.6/tests/IDS/MonitorTest.php:519:          $exploit
s[] = "a//a'\u000aeval(name)";
/var/www/dvwa/external/phpids/0.6/tests/IDS/MonitorTest.php:520:          $exploit
s[] = "a//a';eval(name)";
/var/www/dvwa/external/phpids/0.6/tests/IDS/MonitorTest.php:580:          $exploit
s[] = "if(0){} else eval(new Array + ('eva') + new Array + ('l(n') + new Array +
 ('ame) + new Array') + new Array)
/var/www/dvwa/external/phpids/0.6/tests/IDS/MonitorTest.php:582:          $exploit
s[] = "switch('foo bar foo bar foo bar') {case eval(new Array + ('eva') + new Ar
ray + ('l(n') + new Array + ('ame) + new Array') + new Array):}";
/var/www/dvwa/external/phpids/0.6/tests/IDS/MonitorTest.php:1073:          $exploi
ts[] = 'ACM=1,1+eval(1+name+(+ACM-1),ACM)';
/var/www/dvwa/external/phpids/0.6/tests/IDS/MonitorTest.php:1074:          $exploi
ts[] = '1+eval(1+name+(+1-1),-1)';
```

Tools to Help Finding Shell Backdoor

- **Checking PHP Backdoor / Web Shell / Backdoor Shell in Advanced :**

http://www.shelldetector.com/

http://www.whitefirdesign.com/tools/basic-backdoor-script-finder.html

http://resources.infosecinstitute.com/web-shell-detection/

http://25yearsofprogramming.com/blog/2010/20100315.htm

http://resources.infosecinstitute.com/checking-out-backdoor-shells/

https://bechtsoudis.com/hacking/detect-protect-from-php-backdoor-shells/

IR Step : Recovery

Objective: Restore the system to normal operations.

- **Change all user passwords**, if the web server provides user-authentication, and you have evidence/reasons to think the passwords may have been compromised. This can require a large user communication
- **If backup server has been used, restore the primary web server component as nominal**

IR Step : Recovery (Cont'd..)

Sample Technical Activity for Recovery Phase :

1. Restore from Backup Files
2. Make Sure the Backup Contain no Backdoor
3. Patch Vulnerability from Last Backup

IR Step : Lesson Learned

Objective: Document the incident's details, discuss lessons learned, and adjust plans and defences.

Communication

- If the defacement has been visible for part of your users, plan to explain the incident publicly.

Report

- A crisis report should be written and made available to all of the involved parties.

IR Step : Lesson Learned (Cont'd..)

The following themes should be described:

- Initial detection;
- Actions and timelines;
- What went right;
- What went wrong;
- Incident cost.

In case of vulnerability discovery, **report any undocumented vulnerability** lying on a product running on the web server (like a PHP forum) to its editor, so that the code can be upgraded in order to release a fix.

What to Do After IR Step Done

- Hardening
  - Hardening the Infrastructure (Web Server, DB Server)
  - Hardening the Web App (Source Code Review, Penetration Testing)
- Implementation Web Application Firewall
- Implementation IDS / IPS
- Implementation File Integrity Monitoring
- Implementation Patch Management Program

Hardening (1)

- **Hardening the Infrastructure**

- Operating System Hardening Reference :
  - Linux, Windows Server, Solaris Server Hardening : https://www.cisecurity.org/cis-benchmarks/
- Web Server Hardening :
  - Apache Hardening Reference : https://www.cisecurity.org/benchmark/apache_http_server/
  - Nginx Hardening Reference : https://geekflare.com/nginx-webserver-security-hardening-guide/
  - IIS Hardening Reference : https://www.cisecurity.org/benchmark/microsoft_iis/

Hardening (2)

- **Hardening the Infrastructure**

❖ DB Server Hardening :

- ➢ MySQL Hardening Reference : https://www.cisecurity.org/benchmark/oracle_mysql/
- ➢ MS SQL Hardening Reference : https://www.cisecurity.org/benchmark/microsoft_sql_server/
- ➢ PostgreSQL Hardening Reference : https://www.cisecurity.org/benchmark/postgresql/
- ➢ Oracle hardening Reference : https://www.cisecurity.org/benchmark/oracle_database/

Hardening (3)

- **Hardening the Web Apps**

❖ Conducting Regular Security Assessment (Penetration Testing, Vulnerability Assessment)

❖ Perform Source Code Review Analysis (Static and Dynamic Analysis)

Tools Reference :

https://www.owasp.org/index.php/Source_Code_Analysis_Tools

https://github.com/mre/awesome-static-analysis

❖ Employ Secure SDLC Model in Web Apps Development

https://www.owasp.org/index.php/Secure_SDLC_Cheat_Sheet

https://resources.infosecinstitute.com/intro-secure-software-development-life-cycle/

Implementation Web Application Firewall

- **Notable Open Source WAF :**
  a. Apache ModSecurity
  b. Nginx NAXSI
  c. AQTronix WebKnight (Microsoft IIS Platform)
  d. Vulture Project : https://www.vultureproject.org/
  e. Lua-Resty-Waf (See CDEF Magazine : 2nd Edition - https://cdef.id/2nd-edition-bulletin-released/ Tutorial Building Lua-Resty-Waf in Bahasa Indonesia by Cyber Defense Community Member Rendra Perdana)

Implementation IDS / IPS

- **Notable Open Source IDS / IPS :**
  a. SNORT
  b. Suricata
  c. BRO IDS

Implementation File Integrity Monitoring

Q : Why FIM?

A : To Monitor Changes in Your Web Apps, especially in Homepage to monitor defacement / changes from unauthorized user.

Notable Open Source File Integrity Monitoring :

- OSSEC

https://ossec-docs.readthedocs.io/en/latest/manual/syscheck/

- Wazuh

https://documentation.wazuh.com/3.x/user-manual/capabilities/file-integrity/index.html

- SAMHAIN

https://www.la-samhna.de/samhain/MANUAL-2_4.pdf

Implementation Patch Management Program (1)

- Always subscribe with your vendor security advisory for information patch regarding specific issue within your infrastructure :

Example :

https://www.securityfocus.com/

https://nvd.nist.gov/

https://secuniaresearch.flexerasoftware.com/community/advisories/

Implementation Patch Management Program (2)

- **Define All Asset and Inventory** in Your Organization
- **Prioritize Each of Asset** based on **Criticality** and / or the **Business Needs**
- Create **Patch Management Program** for Your Company
- **Define the Strategy for Patch Management** for Every Infrastructure in your Organization (e.g : **Testing Procedure, Roll Back Procedure, Testing Environment, Documentation of SOP**)
- **Employ Patch Management Technology** to Help you Speed up the Process.

FINISH

Q&A