11. Penetration Testing

Simona Buchovecká, Tomáš Čejka

Faculty of Information Technology, CTU in Prague

simona.buchovecka@fit.cvut.cz, tomas.cejka@fit.cvut.cz

December 7, 2020

Vulnerability / Red Team Assessment, Penetration Test

- **Vulnerability assessment**
  - Using **automated** tools to identify known vulnerabilities in the network
  - Only one phase of penetration testing itself! (reconnaissance)
    - Non-exploitable vulnerabilities, False positives
    - Patched != Secure (Design vulnerabilities)
- **Penetration test**
  - Identify vulnerabilities and try to exploit them to penetrate a system/network to detect if the vulnerability is genuine
  - Usually **manual** effort involved
- **Red team assessment**
  - Main goal: to test organization's detection and response mechanisms & processes

Phases of Penetration Test #1

1. Pre-engagement

Scope, testing window, contact information, written approvals, NDAs

2. Information Gathering

Passive Reconnaissance (OSINT, sniffing), Active Reconnaissance (social engineering, scanning)

3. Threat Modeling

Think like attacker, develop scenario

4. Vulnerability Analysis

Vulnerability scanners (Qualys, Nessus, OpenVAS)

Phases of Penetration Test #2

5. Exploitation

Run the exploits, attempt to access client's system

6. Post-exploitation

Gather the information about the system, look for interesting files, attempt to elevate privileges...

7. Reporting

Executive Summary (background, overall posture, risk profile, general findings, recommendations summary), Technical report (all the details)

Testing Types

- Black box vs. White box vs. Gray box
- Inside attacks vs outside attacks
- Active vs. Passive attacks

Entry Points

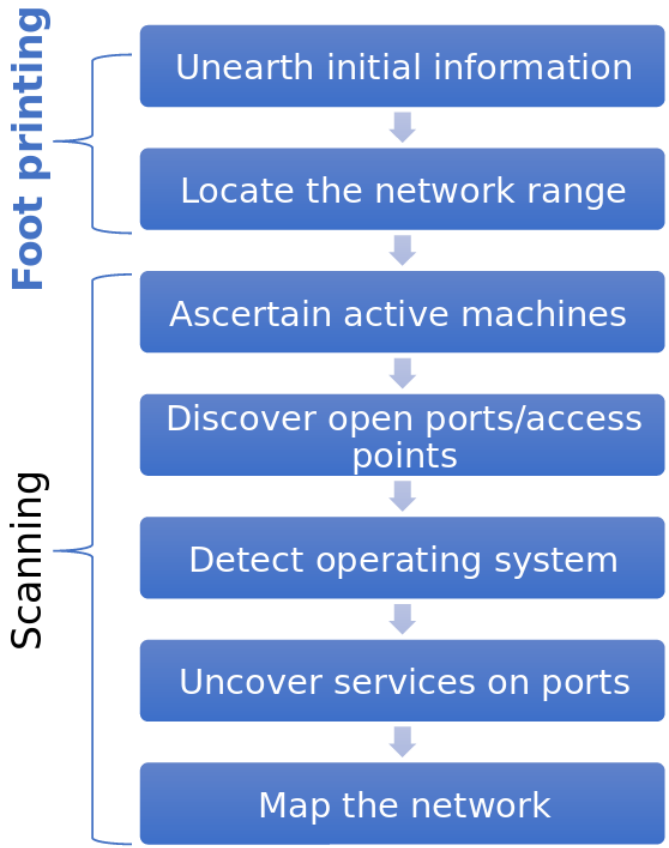
- Remote Network — to simulate an attack launched over the Internet
  - Break a vulnerability in the outside defenses of the network — FW, proxy, router, ...
- Local Network — to simulate someone with physical access to the network and gaining additional unauthorized access, WLANs fall into this category (attacker can stay out of the building)
- Stolen equipment
- Social engineering
- Physical entry

Reconaissance

- Term coming for military — active seeking an enemy's intention by collecting and gathering information about an enemy's composition and capabilities via direct observation
- Goals
  - Gather as much information as possible
  - Create list of attackable IP addresses out of this info
- Active vs. Passive

Information Gathering Methodology #1

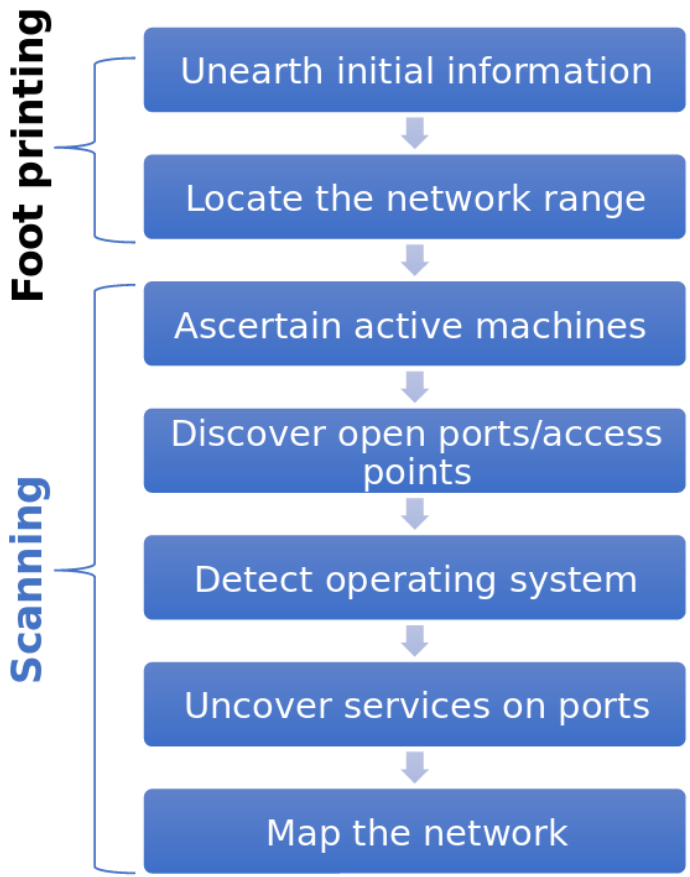
Similar processes/techniques, as the attacker use (and we discussed through the course)



- Web servers
- Whois, URL analytics
- Search engines, webarchives
- Google Earth
- Employee sites, customers
- Financial webs
- Job sites
- Patent/trademarks, Press releases
- Dumpster diving, shoulder surfing, eavesdropping
- Social engineering

Information Gathering Methodology #2

Similar processes/techniques, as the attacker use (and we discussed through the course)



- Input from foot printing
  - gathered information
  - understanding target
  - list of IPs (that we are authorized to attack)
- Scanning
  - Determine if the system is alive (unreliable, should always continue with next steps)
  - Port scan (to identify the specific ports and services running on the particular host)

Practice your Google-Fu

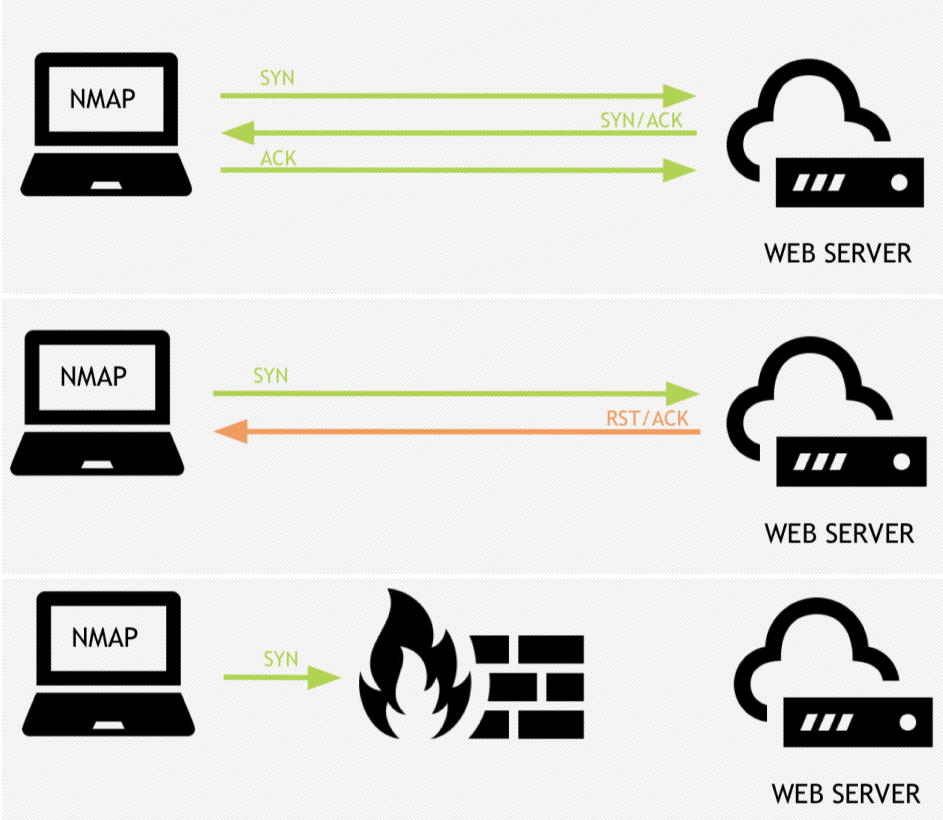
- Useful Google directives
  - site:domain
  - allintitle:, intitle:
  - inurl:
  - filetype:
- Google cache — cache:
- Considered passive only until clicking a link
- Google Hacking for Penetration Testers, Johnny Long's book & DEFCON presentation (`https://www.youtube.com/watch?v=fo1BR9itwOY`)

Detecting Ping Sweeps & Port Scans

- Almost all IDS/IPS will detect and alert on ping sweep/port scan occurring on network
- Most firewalls and proxy servers block ping responses
- Just because a ping sweep does not return any active hosts on network, it does not mean they are not available

TCP Scanning

- TCP handshake
- Closed ports
- Filtered ports



UDP Scanning

- UDP — no confirmation of received data is sent
- UDP packet sent to an open port
  - response: no response
- UDP packet sent to a port that is not open
  - response: ICMP port unreachable
- Absence of response → considering port open
- If port blocked on firewall, false consideration of port being open
- Slow (longer timeout) & unreliable

Banner Grabbing and OS Fingerprinting

- Banner Grabbing
  - process of opening a connection and reading a banner or response sent by application
  - many applications such as email, FTP, web servers will respond to a telnet connection with the name and version of the software
- OS Fingerprinting
  - Active: sending a data to a system to see how it responds, based on the fact that different vendors implement the TCP stack differently and response will differ based on OS
  - Passive: stealthier, examines traffic on the network. Uses sniffing techniques. Usually undetected, but less accurate.

Capturing the Traffic

- Insider threat or an attacker who has breached the perimeter simulation
- Capturing the traffic from various systems in the networks can provide us with interesting info (ultimately usernames and passwords)
- Methods to get the traffic that was not intended for us (you know them from previous lectures):
  - ARP Cache poisoning
  - DNS Cache poisoning
  - Encrypted traffic $\rightarrow$ SSL attacks

Vulnerability scanning

- Vulnerability: Weakness in the software or system configuration that can be exploited
- Vulnerability scanners (Qualys, Nessus)
  - Check if the remote host is alive
  - Firewall detection
  - TCP/UDP port scanning
  - OS detection
  - TCP/UDP Service Discovery
  - Vulnerability assessment based on the services detected
- `cve.mitre.org`: Common Vulnerabilities and Exposures system provides a reference-method for publicly known information-security vilnerabilities and exposures
- or `https://www.exploit-db.com/`

Threat Modeling

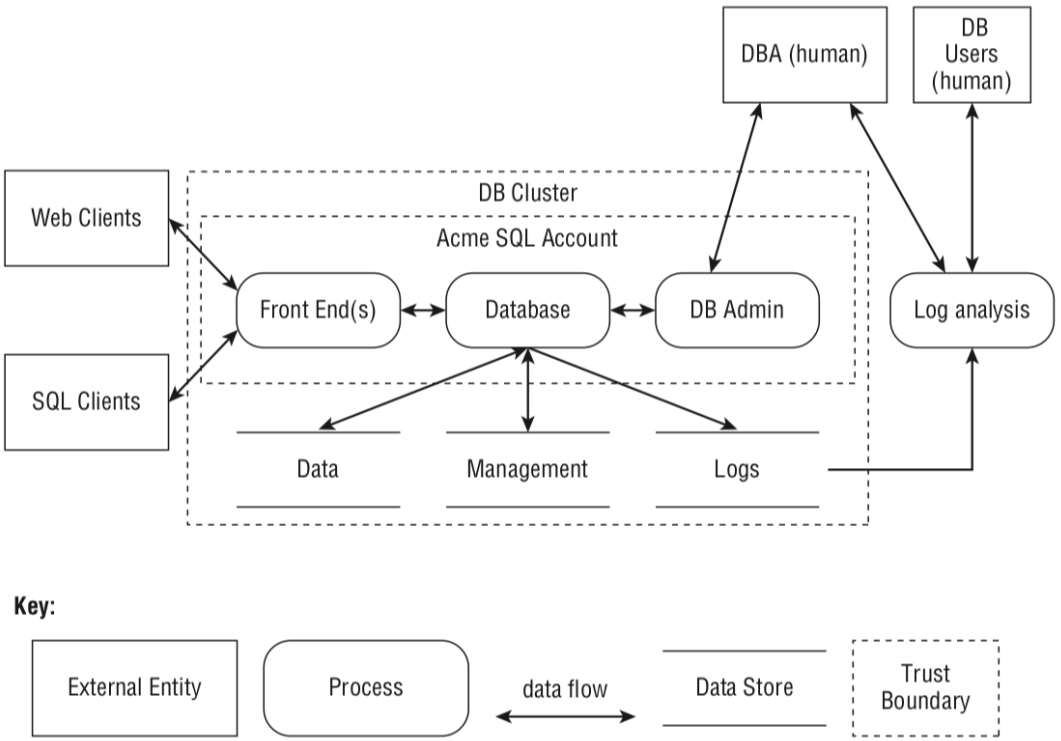
what is threat modeling?

- **analysis** which exposes *possible threat vectors*, leading to better understanding of a system, asset, or attacker for **defensive** purposes
- primary used as a tool to develop defensive countermeasures
- currently focuses on analysis of system, asset or attacker
- “understand the attack” > “design a compensating defense”
- “how will this be attacked?” “where should we fortify defenses?”

src: Offensive Threat Modelling for attackers: `https://media.blackhat.com/bh-eu-12/Los/bh-eu-12-Los-Offensive_Threat_Modeling-Slides.pdf`

Trust Boundaries

Data Flow Diagram (Example)



src: `https://users.encs.concordia.ca/~clark/courses/1601-6150/scribe/L04c.pdf`

Threat Modeling with STRIDE

Threat	Property Violated	Definition	Example
Spoofing	Authentication	Impersonating something or someone else.	Pretending to be any of Bill Gates, Paypal.com or ntdll.dll
Tampering	Integrity	Modifying data or code	Modifying a DLL on disk or DVD, or a packet as it traverses the network
Repudiation	Non-repudiation	Claiming to have not performed an action.	"I didn't send that email," "I didn't modify that file," "I *certainly* didn't visit that web site, dear!"
Information Disclosure	Confidentiality	Exposing information to someone not authorized to see it	Allowing someone to read the Windows source code; publishing a list of customers to a web site.
Denial of Service	Availability	Deny or degrade service to users	Crashing Windows or a web site, sending a packet and absorbing seconds of CPU time, or routing packets into a black hole.
Elevation of Privilege	Authorization	Gain capabilities without proper authorization	Allowing a remote internet user to run commands is the classic example, but going from a limited user to admin is also EoP.

src: `https://users.encs.concordia.ca/~clark/courses/1601-6150/scribe/L04c.pdf`

Spoofing On the Local Machine

Threat Example	What the Attacker Does	Notes/Examples
Spoofing a process	Creates a file before the real process	Then your process relies on it
	Abuses names	Create a version of “sudo” and alter PATH
Spoofing a filename	Creates a file in the local directory	Library, executable or config file
	Creates a link, changes it	Also called ‘race condition’ or TOCTOU
	Creates many files in a target directory	Code can easily create all possible /tmp/foo.random

src: `https://users.encs.concordia.ca/~clark/courses/1601-6150/scribe/L04c.pdf`

Spoofing Over a Network

Threat Example	What the Attacker Does	Notes/Examples
Spoofing a machine	ARP spoofing	
	IP spoofing	
	DNS spoofing	
	DNS compromise	Can be at the TLD, registrar or DNS server
	IP redirection	
Spoofing a person	Take over account	"Stranded in London"
	Set the display name	
Spoofing a role	Declares themselves to be that role	Sometimes opening a special account, setting up a domain/website, other "verifiers"

src: `https://users.encs.concordia.ca/~clark/courses/1601-6150/scribe/L04c.pdf`

Information Disclosure (Processes)

Threat Example	What the Attacker Does	Notes/Examples
Extracts user data	Exploits bugs like SQL injection to read db tables	Can find this by looking to data stores, but here the issue is the process returning data it shouldn't
	Reads error messages	
Extracts machine secrets	Reads error messages	Cannot connect to database 'foo' as user 'sql' with password '&IO*(^&'
	Exploits bugs	"Heartbleed"

src: `https://users.encs.concordia.ca/~clark/courses/1601-6150/scribe/L04c.pdf`

Information Disclosure (Data Stores)

Sub-category	What the Attacker Does
Permissions	Take advantage of missing or inappropriate ACLs
	Take advantage of bad database permissions
	File files protected by obscurity
Security	Find crypto keys on disk or in memory
	Get data from logs/temp files
	Get data from swap files
	See interesting information in filenames/directory names
Network	See data traversing a network
Misc	Obtain device, boot in new OS

src: `https://users.encs.concordia.ca/~clark/courses/1601-6150/scribe/L04c.pdf`

Information Disclosure (Data Flow)

Sub-category	What the Attacker Does
Network	Read data on a network
	Redirects traffics to enable reading data on the network
Metadata	Learns secrets by analyzing traffic
	Learns who talks to whom by watching the DNS
	Learns who talks to whom by analyzing social network information

src: `https://users.encs.concordia.ca/~clark/courses/1601-6150/scribe/L04c.pdf`

Example Threat Tracking Tables

Diagram Element	Threat Type	Threat	Bug ID
Data flow #4, web server to business logic	Tampering	Add orders without payment checks	4553 "Need integrity controls on channel"
	Info disclosure	Payment instruments sent in clear	4554 "need crypto" #PCI

Threat Type	Diagram Element(s)	Threat	Bug ID
Tampering	Web browser	Attacker modifies our JavaScript order checking	4556 "Add order-checking logic to server"
	Data flow #2 from browser to server	Failure to authenticate	4557 "~~Add~~ enforce HTTPS everywhere"

src: `https://users.encs.concordia.ca/~clark/courses/1601-6150/scribe/L04c.pdf`

how offensive threat modeling differs

- turns focus on the *defenders*
- attempts to understand **defenses**, or **defenders**
- provides analysis of the *weaknesses*
- seeks to develop an **offensive** strategy based on analysis
- primarily useful for stealth-mode attackers
- useful for penetration testing, assessments

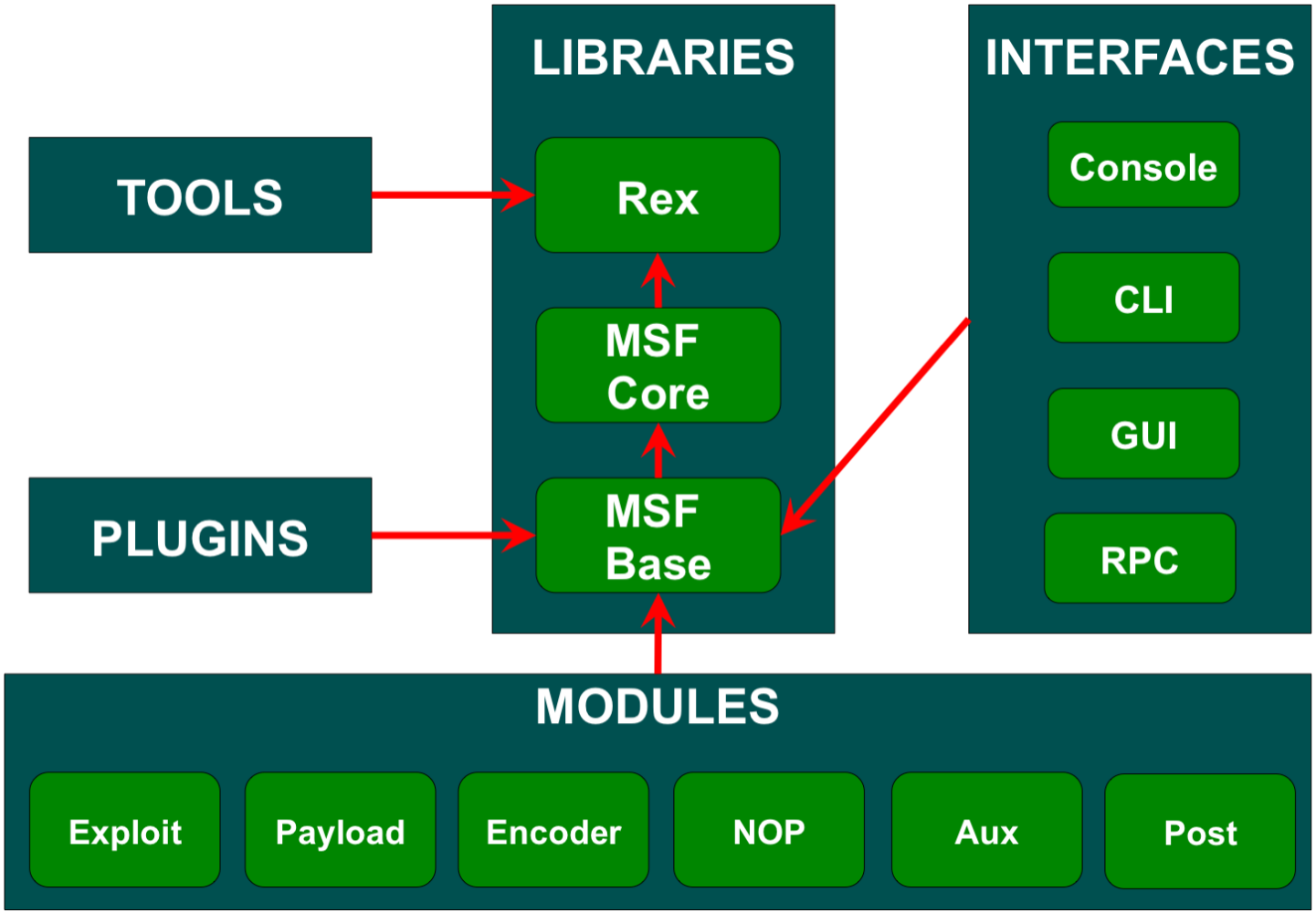
yes … this is how an APT will attack you

src: Offensive Threat Modelling for attackers: `https://media.blackhat.com/bh-eu-12/Los/bh-eu-12-Los-Offensive_Threat_Modeling-Slides.pdf`

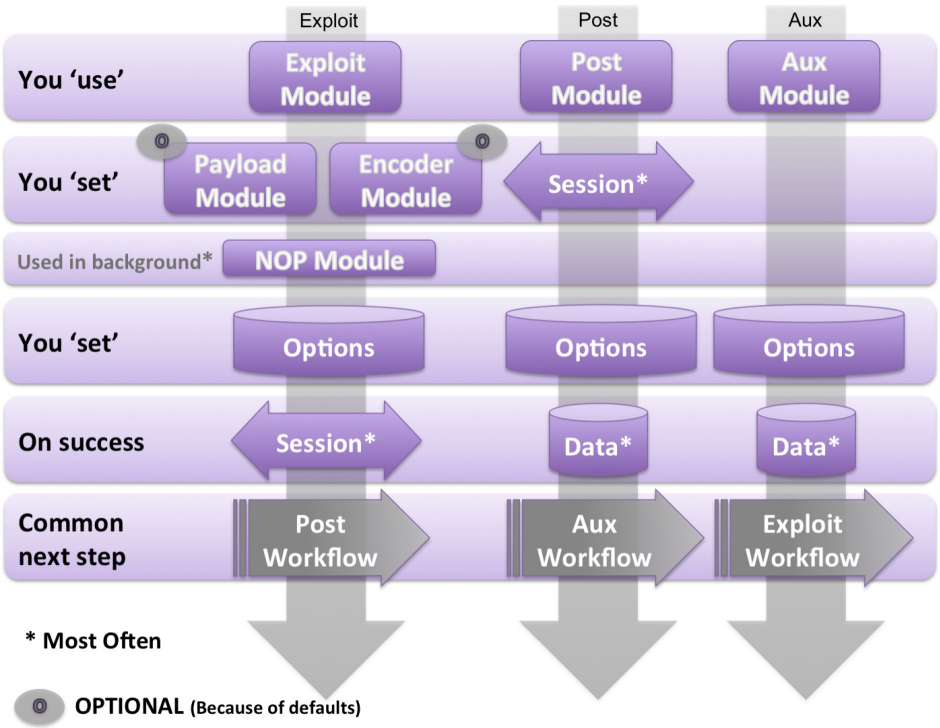
Exploitation

- The exploitation phase of a penetration test focuses solely on establishing access to a system or resource by bypassing security restrictions
- Countermeasures: Host Based Intrusion Prevention System, Security Guard, Web Application Firewall, or other preventative methods.

Metasploit Framework



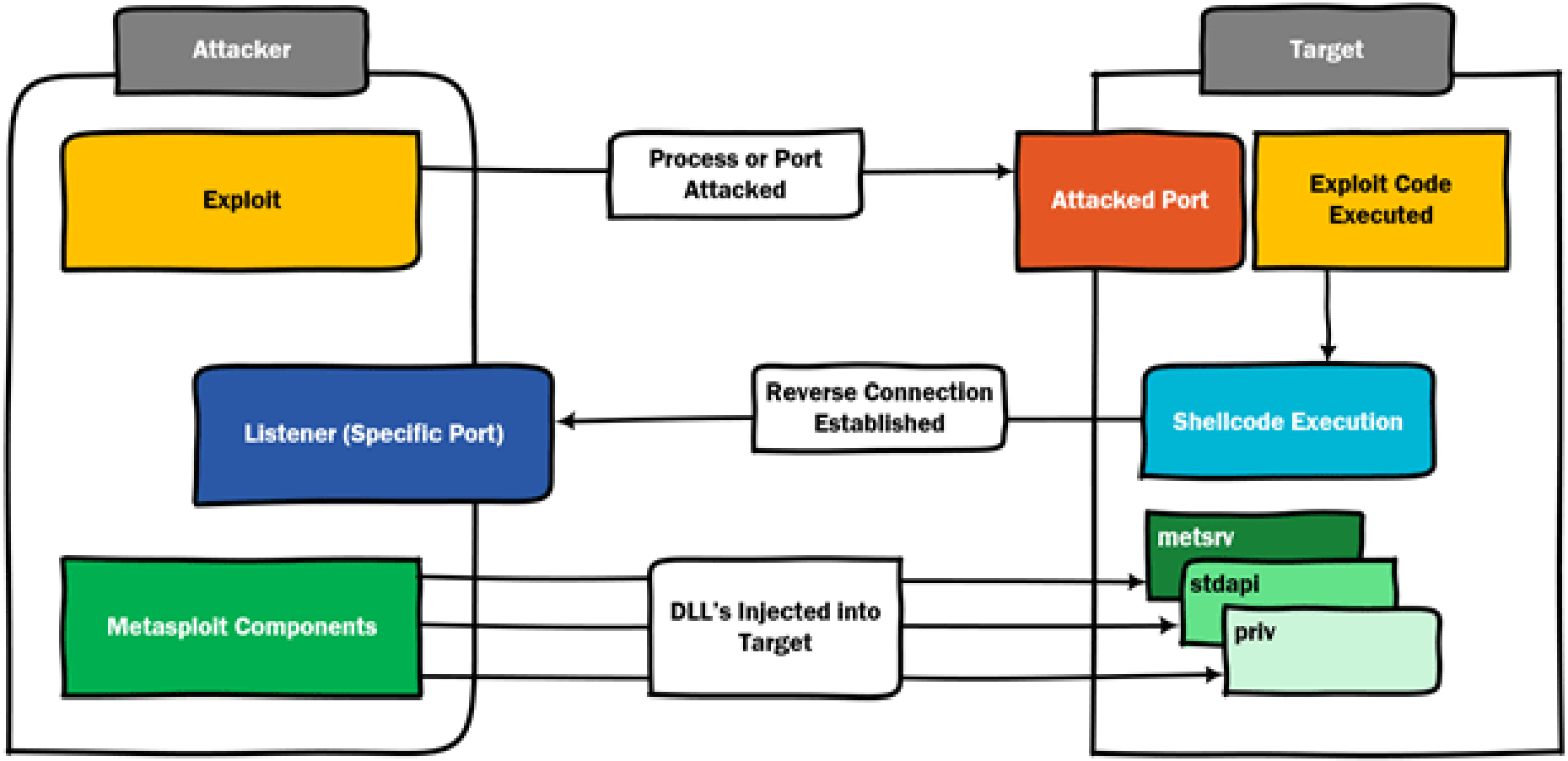
Metasploit Modules



- Payloads
  - singles
  - stagers
  - stages
- Encoders
  - transforming your shellcode into pure alphanumeric, getting rid of bad characters or encoding it for 64 bit target

`https://www.offensive-security.com/metasploit-unleashed/payloads/`

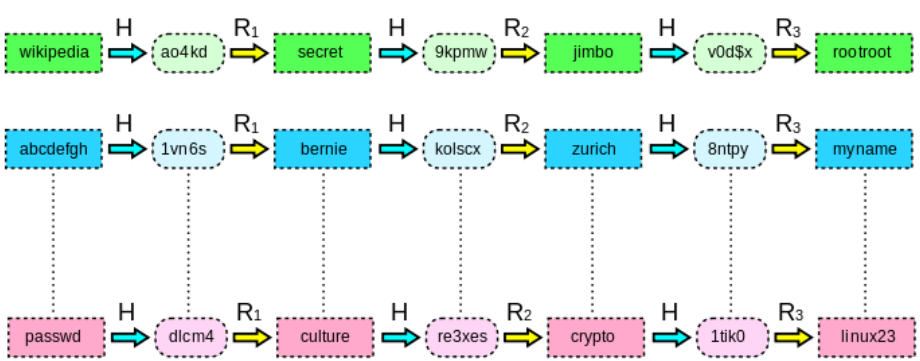
How it Works

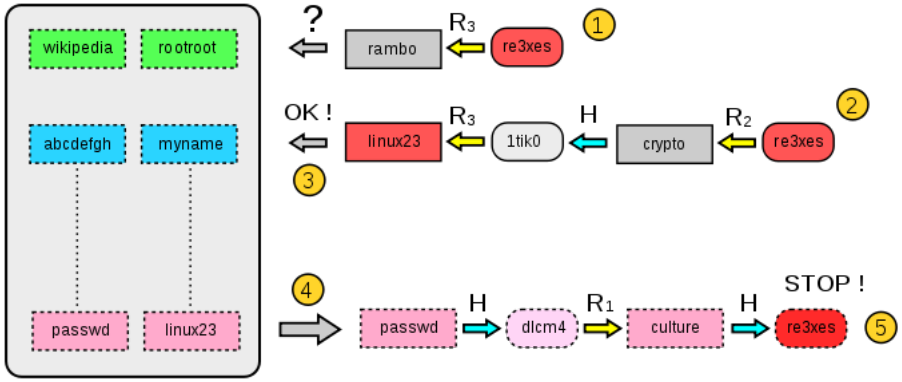

Attacker
Target
Exploit
Process or Port Attacked
Attacked Port
Exploit Code Executed
Listener (Specific Port)
Reverse Connection Established
Shellcode Execution
metsrv
stdapi
priv
Metasploit Components
DLL's Injected into Target

Password Attacks

- Online vs Offline attacks
- Guessing
- Dictionary attacks
- Brute-force attacks
- Rainbow tables

Rainbow Tables





https://en.wikipedia.org/wiki/Rainbow_table#/media/File:Rainbow_table1.svg

https://en.wikipedia.org/wiki/Rainbow_table#/media/File:Rainbow_table2.svg

Fuzzing

- Automated software testing technique that involves providing invalid, unexpected, or random data as inputs to a computer program.
- The program is then monitored for exceptions such as crashes, or failing built-in code assertions or for finding potential memory leaks.

Post Exploitation

- Determine the value of the machine compromised and to maintain control of the machine for later use
- Value of the machine is determined by the sensitivity of the data stored on it and the machines usefulness in further compromising the network

Persistence

- Installation of backdoor that requires authentication.
- Installation and/or modification of services to connect back to system. User and complex password should be used as a minimum; use of certificates or cryptographic keys is preferred where possible. (SSH, ncat, RDP). Reverse connections limited to a single IP may be used.
- Creation of alternate accounts with complex passwords.
- When possible backdoor must survive reboots.

Local Information Gathering

- Infrastructure analysis – network configuration & services, installed services
- Sensitive data
  - key logging, screen capturing
  - network traffic capture
- User information
- System Configuration
- Files

Lateral Movement

- Application Deployment Software
- Logon Scripts
- Pass the Hash
- Pass the ticket
- Remote Desktop
- Remote Copy File
- Remote services
- Windows Admin Shares
- PSExec

Cleanup

- Remove all executables, scripts and temporary file from a compromised system. If possible use secure delete method for removing the files and folders.
- Return to original values system settings and application configuration parameters if they where modified during the assessment.
- Remove all backdoors and/or rootkits installed.
- Remove any user accounts created for connecting back to compromise systems.

References

- `http://www.pentest-standard.org/`
- Kimberly Graves: Certified Ethical Hacker Study Guide
- Georgia Weidman: Penetration Testing

Questions?