Chapter 14

Analyzing Network Traffic

Ed Crowley

Topics

- Finding Network Based Evidence
- Network Analysis Tools
- Ethereal
  - Reassembling Sessions Using Wireshark
- Network Monitoring

Intro

- Once full content data is collected, it needs to be analyzed.
- Purposes/goals include identifying indications and warnings of suspicious activity.

Finding Network Based Evidence

- Methodology should allow a quick drill down and identification of:
  - Relevant network traffic and
  - Potential indicators
- Seeks to confirm that a computer security incident has, or has not, occurred.

Three main steps

1. Identify suspicious network traffic
2. Replay, or reconstruct, suspicious sessions
3. Interpret what occurred

Network Analysis Tools

- TCPTRACE
  - Identifies TCP/UDP sessions within a binary capture file.
  - Unix tool.
- Snort
  - Open source, Intrusion Detection System
- TCPFLOW
  - Reconstructs TCP sessions regardless of retransmission or out of order deliver
- Wireshark (Ethereal)
  - Sniffer capable of viewing the reconstructed streams of a TCP session

Collecting and Reviewing Network Traffic Collected

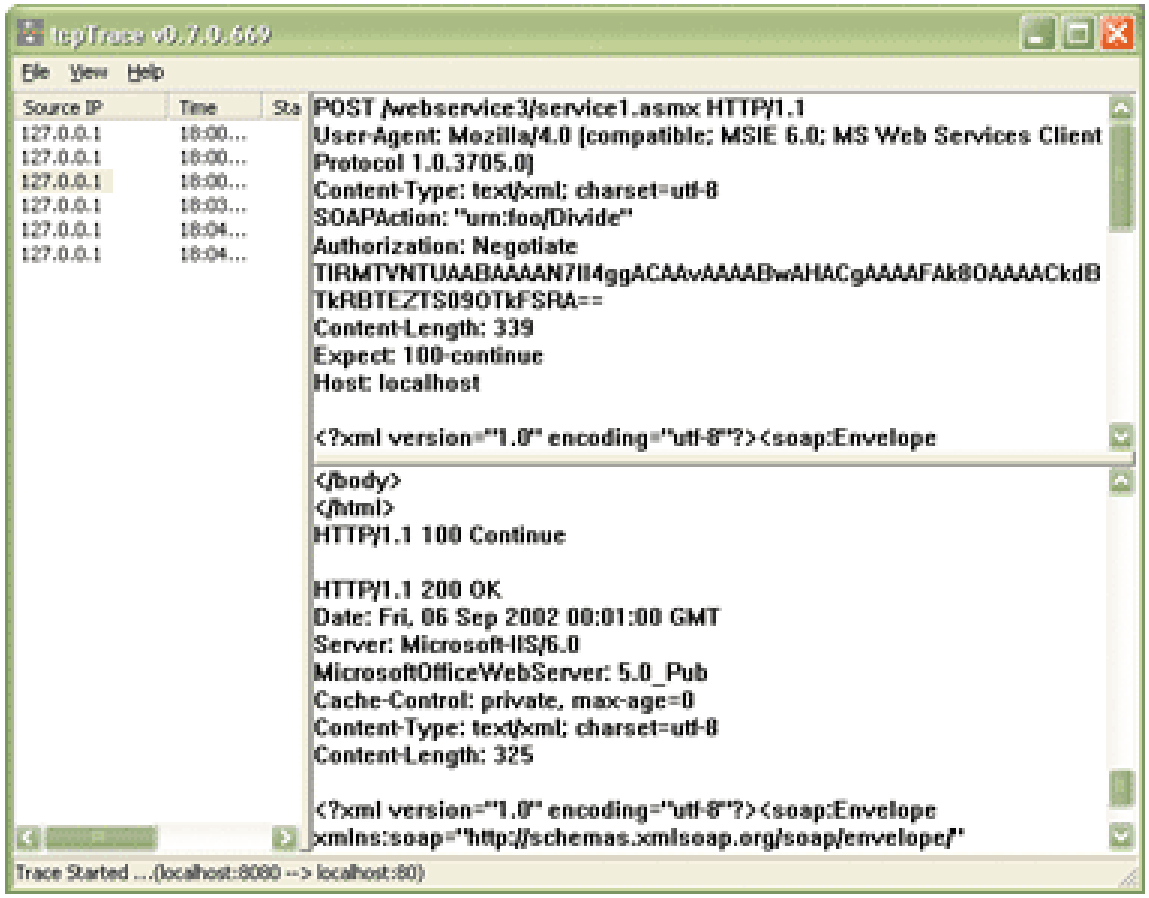
- TCPDUMP in read mode can display packets in a capture file.
  - tcpdump output shows a summary of packets seen on the network.
  - Does not readily present session data.

Other tools, like Wireshark, can help analyze TCP dumps.

Host Resolution

- When analyzing network traffic, it is often useful to not automatically resolve hostnames or port numbers.
  - Resolving the hostname or port number adds overhead through additional tasks without providing any useful information.
    - Potentially slowing down system
    - May overload DNS system
  - If you know the source IP address of a system, you can always determine the hostname later.

Interpreting the tcptrace Output



Using Snort to Extract Event Data

- Snort is an effective way to process large binary capture files

Checking for Syn packets

- Using this rule, we can easily peruse gigabytes of information in a capture file and identify all occurrences of the web server initiating a session to another computer system

Reassembling Sessions Using TCPFlow

- Tcpflow reconstructs data streams and stores each flow in a separate file for later analysis.
  - Understands packet sequence numbers
  - Correctly reconstructs data streams, regardless of retransmissions or out of order delivery.
  - Uses same Berkeley Packet Filter conventions used by tcpdump …
- One of Wireshark's strong suits is that it can replay complete sessions by combining multiple files from tcpflow output.

Reassembling Sessions Using Wireshark

- Wireshark, a GUI based protocol analysis tool that can:
  - Reconstruct TCP sessions
  - Replay both sides of a conversation between hosts
  - Handle IP fragmentation
  - Understand the majority of the known Internet protocols.
  - Replay the TCP stream containing a selected packet.

Note: Your text uses the older name "Ethereal" rather than the name "Wireshark" to refer to the packet analysis tool that we use.

Wireshark

- More user friendly than tcptrace or tcpflow.
- Can color code display data.
  - For example, it can display data sent by the server in blue and data sent by the client in red.

Network Monitoring

- When performing computer intrusion investigations, it is important to place network monitors on the victim network.
  - However, if you can minimize or filter the collected traffic, you likely will more rapidly identify the attacker's methodology.
- To obtain some indicators of the attack,you should perform some host based response (live response) on known compromised computer systems.
- Such analysis may provide you with the knowledge you need to filter to produce relevant traffic.

Questions?

Chapter 15

Investigating Hacker Tools

Ed Crowley

10

Topics

- Tool Analysis Goals
- Compiling and Linking Programs
- Static Analysis of a Hacker Tool
- Reviewing the ASCII and Unicode Strings
- Hex Editors
- Object Code Review
- Performing Online Research
- Using Fport and PsList

Intro

- During computer crime investigations, particularly intrusions, you may encounter rogue files
  - with unknown purposes.
- Here, we present a sound scientific approach to performing tool analysis.

Goals of Tool Analysis

- Prevent similar future attacks
- Assess an attacker's skill and/or threat level
- Determine:
  - Extent of compromise
  - Damage done
  - Number and type of intruders
- Prepare for a successful subject interview (if you catch the attacker)
- Determine attacker's objectives and goals

Compiling Process

- A compiler, such as the GNU C compiler,
  - Reads an entire program written in a high level language, such as C, and
  - Converts it to object code, often called machine code, binary code, or executable code.
- A program can be compiled different ways.
  - Each way affects the amount of information available to the investigator during analysis.

Linking Programs

- A statically linked executable file contains all code necessary to successfully run.
  - Typically, does not have any dependencies.
- Dynamically Linked Programs.
- Nearly all modern operating systems support the use of shared libraries
  - Contain commonly used functions and routines.
  - By compiling a program to use the shared libraries, a programmer can reference them somewhere in memory when the program needs to use those functions and routines, rather than incorporating all that code in the application itself.

Programs Compiled with Debug Options

- Debug compilations are normally used during early stages of program development to help troubleshoot problems and optimize code.
- When debug options are enabled, the compiler will include a lot of information about the program and its source code.

Stripped Programs

- Strip is a function that discards all symbols from the object code to make a file much smaller and perhaps more optimal for execution.
  - Most difficult to analyze.
- UPX, Ultimate Packer for eXecutable
  - Becoming increasingly popular as an effective compression tool for executable files.
- Can also be used to obscure illicit programs from signature based IDS.

Static Analysis of a Hacker Tool

- Tool analysis normally performed without actually executing rogue code.

General examination steps

1. Determine type of file

   Review ASCII and Unicode strings contained within binary file (If any.)

1. Perform online research to determine if the tool is publicly available.
   - Search computer security and/or hacker sites.
   - Compare any online tools identified with the tool you are analyzing.
2. If you have either the source code or believe you have identified the source code via online research, perform source code review.

File

- The standard command for determining a file type on Unix systems is file.
- The Windows equivalent of the file command is the NT Resource Kit tool exectype.

Reviewing ASCII and Unicode Strings

- Basic static analysis of object code involves examining the ASCII formatted strings within the binary file.
- On Windows based executables, it is important to also perform Unicode string searching.
  - Windows 2000 is built on Unicode, and many Windows based application use Unicode.
  - Unicode is a standard character set that uses 2-byte values to represent a character.

Hex Editors

- To computer investigators, hex editors are what hammer and nails are to a carpenter.
- Anything that the program does not dynamically create or take in from another source, such as command line interaction, may be found in the object code.

Object Code Review

Look for the:

- The name of the source code files before the application was compiled.
- Exact compiler used to create the file.
- “Help” strings in the tool
- Error messages that the program displays
- Value of static variables

Performing Online Research

- Perform the strings command on rogue executable files to determine the compiler used to create the executable file.

Dynamic Analysis of a Hacker Tool

- Dynamic tool analysis is when you execute rogue code and interpret its interaction with the host operating system

Includes:

- Monitoring
  - Time/date stamps to determine what files a tool affects
  - How Windows based executables interact with the Registry
- Run the program to intercept the system calls
- Monitor network, determine if any network traffic is generated

Creating the Sandbox

- VMware allows you to run tools in a controlled environment that will not damage the forensic workstation on which you are executing rogue code.
  - Nonpersistent writes, a feature of VMware, allows the investigator to execute rogue code in an environment where the ill effects of the rogue code will not be saved to the disk.
  - Make sure that test system is not an Internet Node
- If you suspect the rogue code may create or respond to network traffic, execute it on a closed network

Dynamic Analysis on a Unix System

- Most applications execute in a memory area defined as user space.
  - User space applications are typically prohibited from directly accessing computer hardware and resources.
  - These resources are controlled by the kernel to enforce security, maintain nonconcurrent use, and provide stability of the operating system.
- The user application makes these request to the kernel via system calls.

Using Strace

- Unix has a tool that traces the use of system calls by an executed process.
- Strace (system trace) is essentially a wiretap between a program and the operating system.
- Displays information about file access, network access, memory access, and many other system calls that a file makes when it is executed.

File Descriptors

- File descriptors are nonnegative integers that the operating system (kernel) uses to reference the files being accessed by a process.
  - File descriptors 0,1, and 2 are the predefined file descriptors for standard input, standard output, and standard error, respectively.
  - When the kernel opens, read, writes, or creates a file or network socket, it returns a file descriptor (integer) that is used to reference the file or network socket.

Using Shortcuts with Strace

- When reviewing strace output, you will be interested in only a few of the system calls, and you will rarely need to be concerned about memory allocation calls such as brk, nmap, and munbmap.
- Authors recommend that you search the strace output file for open, read, write, unlink, lstat, socket, and close system calls.

Sockets and Ports

- Next, determine which sockets are open and which processes are responsible for listening on each socket.
- Linux's `netstat -amp` command maps processes to open ports.

Tools

- Strace utility cannot do everything
- To use decompiliers and debugging techniques, you need to understand the structure of Unix program files.
- The binutils package that is installed on most versions of Linux is built to recognize a small number of object file types.
  - Means that the tools in the precompilied binutils package may build, view, disassemble, and otherwise alter a handful of Linux native executable files.

Dynamic Analysis on a Windows System

Concept

- Execute rogue code.
  - Use utilities to observe how the rogue process interacts with:
  - File system
  - Registry
  - Application programming interfaces (APIs)
  - Operating system.

Utilities

- Filemon
- Regmon
- ListDLLs
- Fport
- PsList

Using Regmon

- Sysinternal's Regmon taps a process's interaction with the Windows Registry.
  - Allows you to enter filters to focus your analysis on relevant entries
  - Provides immediate access to the Registry Editor (regedit)
- Regmon provides a simple interface to monitor which programs write startup entries in the Registry and which programs query the network hardware in order to generate or receive network traffic.

ListDLLs

From the NTRK, ListDLL shows all of the DLLs needed by a process

- Enumerates the full pathnames of the DLLs loaded by the process
- ListDLLs is helpful for detecting applications that have been modified (injected) with extra functions
- For ListDLL to work, the program must be running.

Using Fport and PsList

- Critical tools for dynamic analysis on a Windows system.
- Fport should be used prior to and after executing a rogue process to determine if the rogue process opened any network sockets.
- Fport is used to identify rogue processes opening network sockets.

Tools

- Tools described here provide the first level of analysis.
- Decompiling and debugging are the next steps.

Questions?