Project 1

Adhere to submission guidelines
- Submit all and only the required files
  - .c, .h, Makefile for each folder – supercipher, togglecipher and losscipher
  - Don’t submit project files from Visual Studio or exe files
- Submit PS or PDF of your writeup
- Submit Makefile
- Make sure your project compiles on EECIS Unix machines

For losscipher address both bit loss and block loss

Makefile

File that holds rules for your program compilation

Example:

```make
all: cipher.c functions.c functions.h cipher.h allincludes.h

gcc -o supercipher -Ifunctions.h -Icipher.h -Iallincludes.h functions.c cipher.c
```

It Is Time To Open Emulab Account

Go to http://www.emulab.net

Sign up with existing project CIS662 (unfortunately I asked for a wrong name)

Go to Documentation link and read about how to use Emulab

Intrusions

Why do people break into computers?
What type of people usually breaks into computers?
I thought that this was a security course. Why are we learning about attacks?

Intrusion Scenario

1. Reconnaissance
2. Scanning
3. Gaining access at OS, application or network level
4. Maintaining access
5. Covering tracks

Disclaimer

Some techniques and tools mentioned in this class could be:
- Illegal to use
- Dangerous for others – they can crash machines and clog the network
- Dangerous for you – downloading the attack code you provide attacker with info about your machine

Don’t use any such tools in real networks – especially not on EECIS network
You can only use them in a controlled environment
Phase 1: Reconnaissance
Get a lot of information about intended target:
- Learn how its network is organized
- Learn any specifics about OS and applications running

Low Tech Reconnaissance
Social engineering
Instruct the employees not to divulge sensitive information on the phone
Physical break-in
Insist on using badges for access, everyone must have a badge, lock sensitive equipment
How about wireless access?
Dumpster diving
Shred important documents

Web Reconnaissance
Search organization’s web site
Make sure not to post anything sensitive
Search information on Usenet postings
Instruct your employees what info should not be posted
Find out what is posted about you
Use Google to find all documents mentioning this company to find out partner companies
Find out what is posted about you

Whois databases
When an organization acquires domain name it provides information to a registrar
Looking at public registrar files one can find out:
- Registered domain names
- Domain name servers
- Contact people names, phone numbers, E-mail addresses

ARIN databases
Find out range of IP addresses assigned to a company
This will be useful later for scanning
http://www.arin.net/whois/arinwhois.html
ARIN databases
OrgName: University of Delaware
OrgID: UNIVER-19
Address: 192 South Chapel Street
City: Newark
StateProv: DE
PostalCode: 19716
Country: US
NetRange: 199.75.219.0 - 199.75.219.255
CIDR: 199.75.219/24
NetName: NET-199-75-219-0-1
NetType: Reassigned
Parent: NET-199-75-0-0-1
Comment:
RegDate: 1996-10-18
Updated: 1996-10-18
TechHandle: RWR3-ARIN
TechName: Reisor, Ron W.
TechPhone: +1-302-831-6030
TechEmail:

Domain Name System
What does DNS do?
How does DNS work?
Types of information an attacker can gather:
- Range of addresses used
- Address of a mail server
- Address of a web server
- OS information
Comments

Interrogating DNS – Zone Transfer
$ nslookup
Default server: evil.attacker.com
Address: 10.11.12.13
server 1.2.3.4
Default server: dns.victimsite.com
Address: 1.2.3.4
set type=any
ls -d victimsite.com
system1 1DINA 1.2.2.1
1DINHINFO "Solaris 2.6 Mailserv" 1DINMX 10 mail1
web 1DINA 1.2.11.27
1DINHINFO "NT4www"

Protecting DNS
Provide only necessary information – no OS info and no comments
Restrict zone transfers – allow only a few necessary hosts
Use split-DNS

Split-DNS
Show different view to external and internal users

Reconnaissance Tools
Tools that integrate ping, whois, ARIN, DNS interrogation and many more services:
Applications
http://www.samspade.org/ssw
Web based portals
http://nettool.false.net
http://www.samspade.org
http://members.tripod.com/mixtersecurity/evil.html
http://www.network-tools.com
Phase 2: Scanning
Detecting information useful for break-in
- Live machines
- Network topology
- Firewall configuration
- Applications and OS types
- Vulnerabilities

War Dialing
Finding modem access
Why modems?
- Networks are protected by a firewall, modems punch holes in firewalls
- Modem access may not even be password-protected

War Dialing
Find out several phone numbers to feed into a war dialer
- It will try ranges surrounding them
  - Randomly
  - With random pause intervals
- It will record every success, move on if it encounters busy tone or a human picks up
- It takes about an hour to check 100 numbers

War Dialing Tools
THC-Scan
- Windows Application
- http://thc.inferno.tusculum.edu
- Easy to use interface
- Automatic but accepts user input
TBA
- PDA application
- http://www.l0pht.com

After War Dialing
Gain access by guessing passwords
Gain information about OS
- If modem sends a string of characters identifying server application, use specific application client to access it

Defenses Against War Dialing
Do not allow users to install modems
- Dial-out modems only
- Find your modems before the attackers do
Network Mapping
Finding live hosts
Ping sweep
TCP SYN
Map network topology
Traceroute
Sends out ICMP or UDP packets with increasing TTL
Gets back ICMP_Time_Expired message from intermediate routers

Traceroute Example
traceroute to copland.udel.edu (128.175.13.92), 30 hops max, 38 byte packets
1 edward (131.179.192.1) 0.278 ms 0.288 ms 0.288 ms
2 131.179.187.3 (131.179.187.3) 0.412 ms 0.431 ms 0.431 ms
3 compsci–mathsci.backbone.udel.edu (131.179.121.1) 0.704 ms 0.809 ms 0.705 ms
4 mathsci–core.backbone.udel.edu (131.179.10.10) 0.815 ms 0.849 ms 0.816 ms
5 core–border.backbone.udel.edu (131.179.16.1) 1.677 ms 1.677 ms 1.571 ms
6 dsl-lax-c2-tms1-p1.cs.ucla.edu (137.164.22.42) 2.010 ms 1.803 ms 2.138 ms
7 bpr-lax-bpr-sl-cl-p2-g2.cs.ucla.edu (137.164.22.21) 2.872 ms 2.383 ms 2.872 ms
8 abilene-la–bpr-lax-p1–15ge.cenic.net (137.164.25.3) 16.122 ms 10.820 ms 10.820 ms
9 hstnng-losang.abilene.ucaid.edu (198.32.8.22) 33.603 ms 33.774 ms 34.025 ms
10 atlang-hstnng.abilene.ucaid.edu (198.32.8.34) 46.541 ms 46.815 ms 46.587 ms
11 washng-atla.abilene.ucaid.edu (198.32.8.66) 73.733 ms 73.866 ms 73.957 ms
12 chp-br4-p-0-0-0.nss.udel.edu (128.175.137.9) 77.319 ms 77.861 ms 77.861 ms
13 chp-rt2-u-9.nss.udel.edu (128.175.111.198) 77.515 ms 77.472 ms 77.515 ms
14 copland.udel.edu (128.175.13.92) 77.281 ms 77.266 ms 77.266 ms

Network Mapping Tools
Cheops
Linux application
http://www.marko.net/cheops
Automatically performs ping sweep and network mapping and displays results in GUI

Defenses Against Network Mapping
Filter out outgoing ICMP traffic
Maybe allow for your ISP only
Use NAT

Port Scanning
Finding applications that listen on ports
Send various packets:
Establish and tear down TCP connection
Half-open and tear down TCP connection
Send invalid TCP packets: FIN, NULL, Xmas scan
Send TCP ACK packets – find firewall holes
Obscure the source – FTP bounce scans
UDP scans
Find RPC applications

Port Scanning
Set source port and address
To allow packets to pass through the firewall
To hide your source address
Use TCP fingerprinting to find out OS type
TCP standard does not specify how to handle invalid packets
Implementations wildly differ
Port Scanning Tools

Nmap
Unix and Windows NT application and GUI
http://www.insecure.org/Nmap
Various scan types
Adjustable timing

Defenses Against Port Scanning

Close all unused ports
Remove all unnecessary services
Filter out all unnecessary traffic
Find openings before the attackers do
Use smart filtering

Firewalk: Determining Firewall Rules

Find out firewall rules for new connections
We don’t care about target machine, just about packet types that can get through the firewall
Find out distance to firewall using traceroute
Ping arbitrary destination setting TTL=distance+1
If you receive ICMP_Time_Exceeded packet went through

Defenses Against Firewalking

Filter out outgoing ICMP traffic
Use firewall proxies

Vulnerability Scanning

The attacker knows OS and applications installed on live hosts
He can now find for each combination
Vulnerability exploits
Common configuration errors
Default configuration
Vulnerability scanning tool uses a database of known vulnerabilities to formulate packets and send them to hosts
Vulnerability scanning is also used for sysadmin

Vulnerability Scanning Tools

SARA
http://www.arc.com/sara
SAINT
http://www.wwdsi.com/saint
VLAD
http://razor.bindview.com/tools
Nessus
http://www.nessus.org
Defenses Against Vulnerability Scanning

Close your ports and keep systems patched
Find your vulnerabilities before the attackers do

Phase 3: Gaining Access

Exploit vulnerabilities
Exploits for a specific vulnerability can be downloaded from hacker sites
Skilled hackers write new exploits

Stack-based Overflow Attacks

Stack stores important data on procedure call

Consider a function
void sample_function(char* s)
{
    char buffer[10];
s    strcpy(buffer, string);
    return;
}

And a main program
void main()
{
    int i;
    char buffer[200];
    for(i=0; i<200;i++) buffer[i]='A';
    sample_function(buffer);
    return;
}

Large input will be stored on the stack, overwriting information

Attacker overwrites return address to point somewhere else
‘Local variables’ portion of the stack
Places attack code in machine language at that portion
Since it is difficult to know exact address of the portion, pads attack code with NOPs before and after
Stack-based Overflow Attacks

IDS could look for sequence of NOPs to spot buffer overflows
Attacker uses polymorphism: he transforms the code so that NOP is changed into some other command that does the same thing, e.g. MV R1, R1
Attacker XORs important commands with a key
Attacker places XOR command and the key just before the encrypted attack code, for decryption
XOR command is also obscured

What type of commands does the attacker execute?
Commands that help him gain access to the machine
Writes a string into inetd.conf file to start shell application listening on a port, then uses Netcat to make raw interactive connection to the port
Starts TFTP to transfer Netcat onto the victim, then accesses it
Starts Xterm

How does an attacker discover stack-based overflow?
Looks at the source code
Runs application on his machine, tries to supply long inputs and looks at system registers
Read more at http://packetstormsecurity.nl/docs/hack/smashstack.txt

For system administrators:
Apply patches, keep systems up-to-date
Disable execution from the stack
Monitor writes on the stack
Store return address somewhere else
Monitor outgoing traffic

For software designers:
Apply checks for buffer overflows
Use safe functions

Password Attacks
Attacker attempts to login with some known username, and to guess a password
Trying dictionary words
Trying combinations of dictionary words
Performing brute-force search
Attacker steals encrypted or hashed password file and tries to decrypt it

Make strong passwords
Think of a phrase, take first letters, mix big caps and special characters
Use password filtering software
Use strong encryption/hash techniques
Web Application Attacks

Account harvesting
Gather usernames by observing error messages, then try to guess passwords
Defense: use same error messages for everything

Hijack a session ID
Observe session ID and how it changes between sessions
Change your session ID to another one
Defense: digitally sign or hash session ID, make them long enough and apply timestamps

SQL Piggybacking
Malformed input into Web form may trigger informative message from an SQL server
Input: '111111111'
Error in SQL syntax near '111111111' at line 1
SELECT * FROM account WHERE (userid='10001' and number='111111111')
Attacker then adds SQL commands into input
Input: '111111111'+or+userid%3d'10002'
SELECT * FROM account WHERE (userid='10001' and number='111111111' or userid='10002')
Defense: filter user input